

All copyrights of this article are held by the author/s

To cite this publication

we suggest to use the APA (American Psychological Association) standard:

[Author's Last Name, Initials, (year), Title of work. Retrieved month, day, year, from source.]

You downloaded this document for free
from the



The NetAcademy is an internet research platform organizing the accumulation, dissemination and review of scientific research and publications world wide.

For any use of this document which is not strictly private, scholarly work, please contact the NetAcademy editors respectively the MCM institute at the below address.

The NetAcademy ®
www.netacademy.org
NA.editors@netacademy.org

The NetAcademy project has been developed at the
Institute for Media and Communications Management MCM
University of St. Gallen
Blumenbergplatz 9
CH-9000 St. Gallen, Switzerland
www.mcm.unisg.ch

Media – Formalization and Architecture

Ulrike Lechner, Beat F. Schmid

Media and Communications Management Institute, University St. Gallen
Müller-Friedberg Strasse 8, CH-9000 St. Gallen, Switzerland
email: {Ulrike.Lechner,Beat.Schmid}@unisg.ch

Abstract. Media are explored as model to envision, to design and to formalize platforms for communities of collaborating human or artificial agents. Seminal to our approach are the media concept and the media reference model. The media concept envisions media as platforms of multi-agent systems and the media reference model determines the main components of a medium and guides its application, e.g., in ECommerce or Knowledge Management. The contributions of this paper are a formalization and an architecture for media. The goal is to facilitate artificial agents to reason about media and to act on media autonomously in performing transactions.

Keywords. Medium, Multi Agent System, Logic, Labelled Deductive System, Rewriting Logic.

1 Motivation and Introduction

The buzzword “New Media” describes a variety of applications in ECommerce, EBusiness or Knowledge Management. Examples for such applications are online shops, Intranets, CSCW or CSCL systems. We observe that those media are designed for human agents not for artificial (software) agents. We observe that, at present, the platforms designed primarily for artificial agents are “closed worlds”, to which agents are confined to¹. Moreover, agents on those platforms as well as the agents on Internet, as e.g., spiders, robots, notification or recommendation services are hardly capable of understanding the media they act on, of reasoning and of autonomously performing significant transactions.

Our approach aims at designing media that facilitate collaboration among artificial and human agents. We follow the notion of a *medium* as developed in sociology: Societies can be defined as “system of places”, where every agent has a place with rights and obligations. Those societies are called media and they bind an agent at a place [16]. Clans, firms, nations, enterprises or marketplaces are examples for media. *Agents* may be humans, software agents, organizational units - any entity that plays a role in exchange and communication. We think of artificial agents to be employed as assistants of humans in accessing applications or as representatives of humans [8,9,6,14]. We design media for agents that may act autonomously on media in performing transactions, as, e.g., in buying

¹ e.g., `maker.media.mit.edu`, `talking-heads.cs1.sony.fr`, `kasbah.media.mit.edu`

or selling goods, in seeking for information and in providing information. The “physical” part of the medium is referred to as platform. Note, that the focus of our design is the medium (not the agent).

Seminal to our approach are two models: (1) the *media concept* [15] that envisions media as platforms for (communities of) agents, and (2) the *media reference model* [16] that captures components of a medium as a platform for agents performing transactions and guides the application of the media model, e.g., in ECommerce and Knowledge Management.

The contributions of this paper are a formalization and an architecture for media. The *formalization* enables artificial agents to reason about the media and about the agents’ liabilities and assets on the medium. The *media architecture* is our general structure for formalizing media as prerequisite for an open society of agents roaming open media - where media descriptions allow artificial agents to learn about media. Formalization and architecture are based on General Logic [2], Labelled Deductive Systems [5] as frameworks and the notation of Rewriting Logic [12] and Modal Logic [10,13]).

Our goals in formalization and architecture are (1) to formalize all aspects of media relevant for an artificial agents to reasoning about the medium, to act and perform transactions on it, and (2) to identify core media components and structures. Note that our goal is not a universal description formalism for media - but a general framework according to which component and application specific formalisms and models can be composed to a medium.

This paper is organized as follows. In Sect. 2, the media concept and its formalization are given. Analogously, Sect. 3 presents the media reference model with formalization and examples. Sect. 4 presents the media architecture. In Sect. 5, related work is discussed and Sect. 6 concludes with a brief discussion. App. A recalls General Logic and Labelled Deductive Systems.

2 The Media Concept

A medium assigns each agent a place with rights and obligations within a society, i.e., within a system of places. We are interested in media for collaborating human and artificial agents. Artificial agents need a formal description of media, to reason about media and to act on those media. In this section, we give the media concept, formalize it and illustrate it with some examples.

2.1 The Media Concept

The media concept envisions media as spheres for communities of agents. *Media* are described in terms of three main components [15]:

1. A *logical space* with syntax and semantics of the information to be available on the platform, i.e., that may be communicated via its channels. This includes information about some domain (worlds), as well as information about the medium with its organization, channel system and the agents.

2. A *system of channels* to distribute information over space and time. Note, that channels resemble a medium considered as a mere carrier of information.
3. An *organizational system* (or organization) to describe with roles the types of its agents, i.e., the behavior expected from agents, and with protocols the interactions of agents via the channel system.

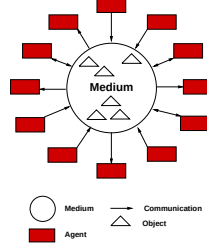


Fig. 1. Medium as Sphere for Communities of Agents

2.2 Formalization of the Media Concept

In formalization, the media concept with its three main components is refined to a media structure with ten components. The frameworks for formalization are General Logic [2,11] and Labelled Deductive Systems (LDS) [5] (See App. A). General Logic is the framework from which languages, theories and models of components are being selected. With LDS, those components are combined and structured to form a medium. Let us give the media structure first and explain it afterwards. We adorn components related to roles by R , protocols by P , channels by C and domain specific components by D .

Definition 1 (Media Structure). Let $L = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$ be a general logic, and $\Sigma_D, \Sigma_C, \Sigma_R, \Sigma_P, \Sigma_{Org}, \Sigma_{Mod}, \Sigma_{Th}$ in $\mathbf{Sign}$ and Th_0 the functor assigning each signature its theories.

A *media structure* is defined as a database by

$$\begin{aligned}
C_R &: L : Th_R : M_R : J_R : n_R : \text{Roles} \\
C_P &: L : Th_P : M_P : J_P : n_P : \text{Protocols} \\
C_D &: L : Th_D : M_D : J_D : n_D : A \\
C_C &: L : Th_C : M_C : J_C : n_C : \text{Channels} \\
C_{Ch} &: L : Th_C : M_C : J_C : \text{ChannelSystem} \\
C_{Org} &: L : Th_{Org} : M_{Org} : \text{OrganizationSystem} \\
C_{Mod} &: L : Th_{Mod} : \text{ModelSystem} \\
C_{Th} &: L : \text{TheorySystem} \\
C_L &: \text{LogicSystem} \\
&MRRoles(C_R), MRProtocols(C_P), MRDomain(C_D), MRChannel(C_C), \\
&MRChannelSystem(C_{Ch}), MROrganizationSystem(C_{Org}) \\
&MRModelSystem(C_{Mod}), MRTheorySystem(C_{Th}), MRLogicSystem(C_L)
\end{aligned}$$

where

$\text{Roles} \subseteq \text{Sen}(\Sigma_R), \text{Protocols} \subseteq \text{Sen}(\Sigma_P), A \subseteq \text{Sen}(\Sigma_D),$
 $\text{Channels}, n_R, n_P, n_C, n_D, \text{ChannelSystem} \subseteq \text{Sen}(\Sigma_C),$
 $\text{OrganizationSystem} \subseteq \text{Sen}(\Sigma_{Org}), \text{ModelSystem} \subseteq \text{Sen}(\Sigma_{Mod}), \text{TheorySystem} \subseteq \text{Sen}(\Sigma_{Th}),$
 $\text{LogicSystem} \subseteq \text{Sen}(\Sigma_L),$ the language of representing a General Logic,
 Th_x in $\text{Th}_0(\Sigma_x), M_x(\text{Th}_x)$ in $\text{Mod}(\text{Th}_x), J_x : |\text{Th}_0(\Sigma_x)| \times |\text{Th}_0(\Sigma_x)| \rightarrow |\text{Th}_0(\Sigma_x)|$
 for $x \in \{R, P, D, C, Org, Mod, Th\}$

The *rows* of a media structure represent components of a medium, more precisely, components of the logical space of information about a medium. We distinguish three different kinds of information, each with several components. We refer to the ten components of a medium with “R” and a number.

Information available on the medium. This includes information about (R1) roles, (R2) protocols, (R3) domain specific information (A) and (R4) information about channels (Channels).

Systems relating distributed information (R5) ChannelSystem models the channel system of a medium. (R6) OrganizationSystem relates the organization (with roles and protocols) on the one hand and domain specific information and channels on the other hand. It captures in how far organization governs interaction on channels and models the binding of an agent to a place. (R7) ModelSystem captures the relations among the models, as e.g., identical interpretation of sub-theories, (R8) TheorySystem captures analogously relations among theories. (R9) LogicSystem captures the relations between logic from which theory and model are being selected.

Media component roles (R10) Predicate (MRRoles . . .) describes which component plays which role in a medium (i.e., provides which information).

The *columns* of the media structure represent each a particular “kind” of information together with a channel relating it. In motivating the columns, we proceed from right to left and assign the columns a number. We distinguish four different groups of information, each with various kinds.

Channel System providing (C1) *explicit knowledge* about roles, protocols, domain and channels and (C2) the *channel system* with names addressing pieces of information and predicate ChannelSystem modeling the channel system of the medium. Note that the distribution of information over time and space on the medium is being captured here.

Organization with (C3) mappings $J_C, J_D, J_P,$ and J_R blending explicit knowledge and implicit knowledge in a component specific way and predicate OrganizationSystem relating organization with channels and agents. (See R6).

Meta-Information with (C4) *Models* which defines in $M_R, M_D, M_P, M_R, M_{Org}$ the semantics for each component and with ModelSystem the relation between the component specific semantics. (C5) The *implicit knowledge* is modeled as theories. The signature of the theory gives information about the language employed in a component. The axioms represent information that is, e.g., prerequisite for understanding the explicit knowledge or that

complements the explicit knowledge, as e.g., information encoded in a machine or service supporting a component. Predicate TheorySystem captures the relations between the theories (see (R8)). (C6) *General Logic* that captures the framework from which language, theory and model of a component are being chosen.

Components (C7) to capture the components of a medium and their role.

Note that all information employed in the media structure is employed in more conventional models as well: Channel systems or platforms as well as agents are designed to implement or satisfy a set of abstract properties - an organization. In our model, the organization and the governing of agents by the platform is explicitly part of the model. This provides more flexibility in the relation agent-channel system to organization than a canonical “|=”.

2.3 Example

A book catalog is modeled as a medium and extended to a bookshop and a library (and continued in Sect. 3.3). As notation, we employ Rewriting Logic [12], temporal and deontic logic [10,13]. The basic propositions of modal logic are sentences of Rewriting Logic, i.e., we have formulas $M\phi$, where M is some modal operator and ϕ some sentence of Rewriting Logic interpreted as a predicate on a state. Note that we employ the Rewriting Logic as our notation and note furthermore that we employ the matching and deduction of Rewriting Logic in modeling. I.e. we employ the matching of Rewriting Logic to match the databases employed rules as sets of literals against a database with sets of declarative units modeling a system. We apply this matching also in the matching of parts of formulas, such that e.g., $m : p$ matches $m, n : p, q$.

The domain of the medium are book descriptions. A book is represented as a tuple $\langle a=A, t=T, n=N, k=K \rangle$ It has attributes author (**a**), title (**k**), ISBD-No. (**n**) and keyword (**k**). **A**, **T**, **N**, **K** represent some attribute values. We assume a lexicographic order on books (**lex**), where **lex**(**X**, **Y**) indicates that **X** is smaller than **Y** w.r.t. **X**.

Channel **next** relates a book with its lexicographic successor. The catalog is based on a channel **bookset** capturing all books.

The roles of the catalog are **Book**, **Firstbook** for books and the **Reader** and **Editor** for human agents or artificial agents.

- (R1) $\text{Reader}(A) = \forall X, Y : \text{next}(X, Y) \Rightarrow \text{Per}(\text{next}(X, Y))\text{tt} \wedge \text{Per}(\text{changeentry}(X, Y))\text{ff}$
- (R2) $\text{Editor}(A) = \forall X, Y : \text{next}(X, Y) \Rightarrow \text{Per}(\text{next}(X, Y))\text{tt} \wedge \text{Per}(\text{changeentry}(X, Y))\text{tt}$
- (R3) $\text{Book}(X) = X : B \Rightarrow [\text{reqinfo}(X, Y)]\text{Obl}(\text{answer}(Y, B))\text{tt}$
- (R4) $\text{Firstbook}(X) = \text{tt}$

An agent playing role **Reader** is always permitted to navigate along **next** but is not permitted to perform a **changeentry**. An **Editor** is permitted to navigate along **next** and to perform a **changeentry**. A **Book** is obliged to answer requests for information **reqinfo** with the all the information about a book. **Firstbook** is only a name for a role with which no actions are being associated with. All

role descriptions have a parameter. The value of the parameter indicates which agent is assigned to this role.

A product catalog of three books where 01 is the first book, 03 its successor and 02 the successor of 03 is modeled as:

```

CR   : L : (ΣR, {R1 – R4}) : MR : JR : 0R : Firstbook(01), Book(01), Book(02), Book(03)
CR   : L : (ΣP, P3)       : MP : JP : 0P : P1, P2
CD1  : L : (ΣD, {})       : MD : JD : 01 : < a=A1, t=T1, n=N1, k=K1 >
CD2  : L : (ΣD, {})       : MD : JD : 02 : < a=A2, t=T3, n=N4, k=K1 >
CD3  : L : (ΣD, {})       : MD : JD : 03 : < a=A3, t=T3, n=N3, k=K3 >
CC   : L : (ΣC, {})       : MC : JC : 0C : next(01,03), next(03,02), bookset({01,02,03})
CCh  : L : (ΣC, {})       : MC : JC : next(01,03), next(03,02)
COrg : L : (ΣOrg, {})     : MC : CatOrganizationSystem
CMod : L : ThMod         : CatModelSystem
CTh  : L : CatTheorySystem
CL   : CatLogicSystem
Catalog

```

where Catalog =

```

MRRoles(CR), MRProtocols(CR), MRDomain(CD1, CD2, CD3), MRChannel(CC),
MRChannelSystem(CCh), MROrganizationSystem(COrg),
MRModelSystem(CMod), MRTheorySystem(CTh), MRLogicSystem(CL)

```

We define with this structure the explicit knowledge about roles to capture only the assignment of roles to pieces of information or agents, i.e. the binding of agents on the medium to a place. **Book(01)** describes that the piece of information with address 01 is playing role **Book**. For brevity, we refer to the role and protocol definitions are referred to with a name (E.g., by R1, P1). The distinction of explicit and implicit knowledge is motivated by (1) the distinction of role assignment which are subject to change with agents entering and leaving a medium vs. roles which are not supposed to change (2) the notion of congruence between the descriptions and machine that enforces it. Note that this machine is then captured by M_R giving an operational semantics of the role descriptions. Role and role description have to be part of the medium to facilitate agent to reason about their assets and liabilities while playing the role. Note that one could think of a model where both assignments and role descriptions are part of the explicit knowledge, where the implicit knowledge is constituted from a different set of axioms and where OrganizationSystem employs the implicit knowledge only in the relation organization vs. agents and channels. This resembles, e.g., the notion of interface implementation in some technology binding the agent and some (differing) role description given to the agent.

Explicit information about the channels **next** and **bookset** is being provided. The channel system itself is constituted by channel **next** only. Predicate $\text{MRDomain}(C_{D1}, C_{D2}, C_{D3})$ captures that three different pieces of information constitute the domain specific information on the medium.

The protocols determine that navigating to the next book is always possible, provided a channel exists. Books do not disappear, i.e., if a book exists, then it

exists after all actions. Any changes of the protocols are possible. I.e.,

$$\begin{aligned}
(P1) \quad & \forall X, Y : \text{next}(X, Y) \Rightarrow \langle \text{next}(X, Y) \rangle \text{tt} \\
(P2) \quad & \forall X : C_R : \dots : r : \text{Book}(X) \Rightarrow [-] C_R : \dots : r : \text{Book}(X) \\
& \quad C_D : \dots : X : B \quad C_D : \dots : X : B \\
& \quad \text{MRRoles}(C_R) \quad \text{MRRoles}(C_R) \\
(P3) \quad & \forall P, \exists P' : C_P : \dots : P \Rightarrow [-] C_P : \dots : P' \\
& \quad \text{MRProtocols}(C_P) \quad \text{MRProtocols}(C_P)
\end{aligned}$$

We define all the mappings J_X to be unions of theories. As models we employ the loose approach. M_X maps a theory to its class of term-generated models [19]. The algebras are transition systems consisting of a term-generated algebra and a labeled transition relation. For the semantics of specifications and a specification language for describing specifications see, e.g., [19]. For the definition of the satisfaction relation for modal and deontic logic, we refer to [10]. For the satisfaction relation of roles we require that there is component participating playing the role, i.e., for the role **Book** we define:

$$\begin{array}{ccc}
M_R(\Sigma_R, C_r : \dots : n_r : \text{Book}(X)) & \models & C_D : \dots : X : \langle a=A, t=T, n=N, k=K \rangle \\
\text{MRRoles}(C_R) & & \text{MRDomain}(C_D)
\end{array}$$

CatTheorySystem defines the names and the predicates to describe relations to be the same throughout Channels, Domain, Roles and protocols. We assume a language for specifications with $\subseteq$ describing inclusion of signatures. CatModelSystem requires an identical interpretation of names and channels i.e., The LogicSystem describes identity among general logic and the Media Roles the components of a catalog.

$$\begin{aligned}
\text{CatTheorySystem} &=_{\text{def}} \Sigma_C \subseteq \Sigma_R, \Sigma_D, \Sigma_P, Ax_R, Ax_P, Ax_D \text{ implies } Ax_C \\
\text{CatModelSystem} &=_{\text{def}} M_R(\text{Th}_R)|_{\Sigma_C} = M_P(\text{Th}_P)|_{\Sigma_C} = M_D(\text{Th}_D)|_{\Sigma_C} = M_C(\text{Th}_C)
\end{aligned}$$

We define the organization system of a catalog to be capable of enforcing roles and protocols. I.e., provided that a state satisfies the roles, a transition may happen iff and only iff it adheres to the protocols and the resulting state adheres to the roles:

$$\begin{array}{ll}
\text{CatOrganizationSystem} = & \\
C_R : L : \text{Th}_R : M_R : J_R : n_R : \text{Roles} & \xrightarrow{\phi} C'_R : L : \text{Th}'_R : M_R : J_R : n'_R : \text{Roles}' \\
C_P : L : \text{Th}_P : M_P : J_P : n_P : \text{Protocols} & C'_P : L : \text{Th}'_P : M_P : J_P : n'_P : \text{Protocols}' \\
C_D : L : \text{Th}_D : M_D : J_D : n_D : A & C'_D : L : \text{Th}'_D : M_D : J_D : n'_D : A' \\
C_C : L : \text{Th}_C : M_C : J_C : n_C : \text{Channels} & C'_C : L : \text{Th}'_C : M_C : J_C : n'_C : \text{Channels}' \\
C_{Ch} : L : \text{Th}_C : M_C : J_C : \text{ChannelSystem} & C'_{Ch} : L : \text{Th}'_C : M_C : J_C : \text{ChannelSystem}' \\
C_{Org} : L : \text{Th}_{Org} : M_{Org} : \text{OrganizationSystem} & C'_{Org} : L : \text{Th}'_{Org} : M_{Org} : \text{OrganizationSystem}' \\
C_{Mod} : L : \text{Th}_{Mod} : \text{ModelSystem} & C'_{Mod} : L : \text{Th}'_{Mod} : \text{ModelSystem}' \\
C_{Th} : L : \text{TheorySystem} & C'_{Th} : L : \text{TheorySystem}' \\
C_L : \text{LogicSystem} & C'_L : \text{LogicSystem}' \\
\text{MRRoles}(C_R), \dots & \text{MRRoles}(C'_R), \dots
\end{array}$$

$$\begin{aligned}
&\text{if } M_R(J_R(\text{Th}_R, (\Sigma_R, \text{Roles}))) \models n_D : A \\
&\quad \quad \quad n_C : \text{Channels} \\
&\text{and } M_R(J_R(\text{Th}_R, (\Sigma_R, \text{Roles}))) \models n_D : A \xrightarrow{\phi} n'_D : A \\
&\quad \quad \quad n_C : \text{Channels} \quad n'_C : \text{Channels}' \\
&\text{and } M_P(J_P(\text{Th}_P, (\Sigma_P, \text{Protocols}))) \models n_D : A \xrightarrow{\phi} n'_D : A \\
&\quad \quad \quad n_C : \text{Channels} \quad n'_C : \text{Channels}' \\
&\text{and } M_P(J_P(\text{Th}_P, (\Sigma_P, \text{Protocols}))) \models n_P : \text{Protocols} \xrightarrow{\phi} n'_P : \text{Protocols}' \\
&\text{and } M_R(J_R(\text{Th}'_R, (\Sigma_R, \text{Roles}')))) \models n'_D : A' \\
&\quad \quad \quad n'_C : \text{Channels}'
\end{aligned}$$

Let us discuss some alternatives in modeling. A catalog like the one above is be adequate for, e.g., an online bookshop, where all the roles have to be enforced. We assume that an information system of a library typically has not the power to enforce organization, i.e., to force humans to return overdue books. Let us give the medium structure for the library first and explain it afterwards.

```

CA :L:      ThA      : MA : JR : A1 :Obl(A1, return(B))
CR1 :L:(ΣR1, {R1 – R4}): MR1 : JR1 : OR1 :Firstbook(01),Book(01),Book(02),Book(03)
CR2 :L:      (ΣR2, {}) : MR2 : JR2 : OR2 :Overdue(A1,01),Overdue(X,Y) = Obl(X, return(Y))tt
CP1 :L:      (ΣP1, P3) : MP1 : JP1 : OP1 :P1,P2
CP2 :L:      ThP2      : MP2 : JP : OP2 :LibP
CD1 :L:      (ΣD, {}) : MD : JD :01,K1:< a=A1, t=T1, n=N1, k=K1 >
CD2 :L:      (ΣD, {}) : MD : JD :02,K1:< a=A2, t=T3, n=N4, k=K1 >
CD3 :L:      (ΣD, {}) : MD : JD :03,K3:< a=A3, t=T3, n=N3, k=K3 >
CC1 :L:      (ΣC, {}) : MC1 : JC1 : OC :next(01,03),next(03,02),bookset({01,02,03})
CC2 :L:      (ΣC2, {}) : MC2 : JC2 : OC2 :gen(K1,K2),gen(K1,K3)
CCh1 :L:      (ΣC1, {}) : MC1 : JC1 :next(01,03),next(03,02)
CCh2 :L:      (ΣC2, {}) : MC2 : JC2 :gen(K1,K2),gen(K1,K2)
CCh3 :L:      (ΣC3, {}) : MC3 : JC3 :returnchan(A1)
COrg :L:      ThOrg      :MOrg: LibOrganizationSystem
CMod :L:      ThMod      : LibModelSystem
CTh :L: LibTheorySystem
CL : LibLogicSystem
Library

```

The catalog is extended to a library by adding new components to the catalog. Hereby the catalog itself remains largely unaltered. This example illustrates two different kind of extensions. The first extension is the enrichment of the product catalogue with a directory. Hereby, the addresses of books become enriched by keywords and the channel **gen** describes, how keywords are related in generalization relations. One can think of rules describing how to navigate either on next or on gen channels as well combination of the two, e.g., on a catalog for books on some keywords. The second extension constitutes the library. The library specific roles, protocols and channels are all given in separate modules (i.e., rows). OrganizationSystem, TheorySystem and ModelSystem have to define how components of one kind interoperate as well as the interoperability of different kind of information. E.g., whether components of one kind are being considered separately, whether the theories are joined, or whether library specific modules have precedence over catalog specific modules. The semantics of the roles is defined

the same way as the product catalog: an obligation to return a book is violated iff an agent performs an action different from returning this book. However, the agent may perform this action, resulting in a reminder being sent to him. Thus, the organization system records the violation and checks every action with the organization. If an action does not model the organization, the action may still happen.

3 The Media Reference Model

The media reference model (MRM) [16] describes the kind of communications or transactions that take place on a medium, the information to be available on the medium. Our formalization yields components of a medium, i.e. basic requirements a model of a medium has to have. While the media concept considers communication - the MRM enriches and refines communication to transactions on a medium for the creation of economic value.

3.1 The Media Reference Model

The Media Reference Model (MRM) distinguishes four kinds of communication acts in the creation of economic value in its four phases (depicted in Fig. 2). The four views structures in four views how to establish a medium on a given information- and communications infrastructure.

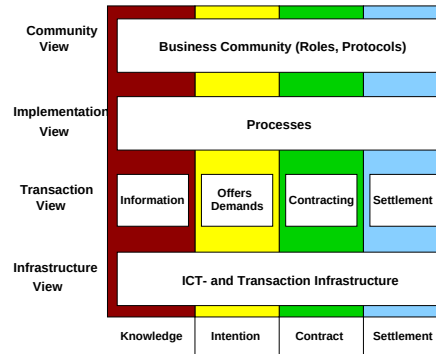


Fig. 2. Media Reference Model (MRM)

Views of the MRM The views relate the channel system, implemented in information- and communication technology with the community's agents and describes how to enrich a channel system to a medium:

The *community view* deals with the aspects relevant for modeling the community of agents, i.e., its organizational structure as roles, protocols, the interests and values behind it, as well as its languages.

The *process view* implements the specified community design, i.e., the community view specifications, as data structures and (business) processes on the services offered by the transaction layer.

The *transaction view* provides generic interaction or communication services, as e.g., signaling of intentions, by supply and demand, or contracting and agreement on contracts and the resulting bindings, or for the performance of the transaction in the settlement of contracts. The services are structured according to the action types agents perform (see below).

The *infra-structure view* provides the means to physically implement the services of the service layer, i.e., process information and performative acts, to transport it over space and time. Here, the respective needs for security, or safety of the infra-structure are being provided.

Phases of the MRM The phases distinguish four kinds of communication acts within the organization [16,7]:

In the *Knowledge Phase*, assertive information about the world, the agents, or the medium is provided and communicated. This logical space includes information about the medium with all its components and its actual state.

In the *Intention Phase*, agents signal their intentions, developed from the knowledge provided in the knowledge phase, and from their desires and goals and by linguistic means of the common logical space, and services provided by the service layer for signaling, following their role and protocols.

In the *Contracting Phase*, agents negotiate contracts. The messages in this phase are binding, in the sense that they oblige agents to act as indicated in those messages. This phase ends - in the case of success- with a contract, i.e., with a protocol that guides the settlement phase.

In the *Settlement Phase*, agents act according the negotiated Contract. In commerce, this means, e.g., shipping of goods and transaction of money.

3.2 Formalization of the Media Reference Model

The MRM describes with the phases kinds of communication acts and with the views the relation between channels as a mere carrier of information to the other components to form a medium. The formalization defines the components a medium description has to have - the media requirements.

Definition 2 (Media Requirements). Let L be a general logic. Let Σ_{TK} , Σ_{TI} , Σ_{TN} , Σ_{TS} of modeling the transitions of knowledge, intention, negotiation and settlement and Σ_T their pushout of the phase specific signatures. Let Σ_R be the language for roles, Σ_P for protocols and Σ_C for channels. A media requirement is defined to be

- (1) $\Sigma_{TK}, \Sigma_{TI}, \Sigma_{TN}, \Sigma_{TS}$
- (2) $\Sigma_R, \Sigma_P, \Sigma_C$,
- (3) $\alpha_R : Sen(\Sigma_T) \rightarrow Sen(\Sigma_R), \alpha_P : Sen(\Sigma_T) \rightarrow Sen(\Sigma_P), \alpha_C : Sen(\Sigma_T) \rightarrow Sen(\Sigma_C)$,

such that

- (M1) $\forall \phi \in \text{Sen}(\Sigma_{\text{TK}}) : I(\phi) \in \{\text{tt}, \text{ff}\}$
- (M2) $\forall \phi \in \text{Sen}(\Sigma_{\text{TI}}) : \text{chroles}(\phi, R) = R \wedge \text{chprotocols}(\phi, P) = P$
- (M3) $\forall \phi \in \text{Sen}(\Sigma_{\text{TN}}) : \text{chroles}(\phi, R) = R \cup_R \text{Role}(\phi) \wedge \text{chprotocols}(\phi, P) = P \cup_P \text{Protocol}(\phi)$
- (M3) $\forall \phi \in \text{Sen}(\Sigma_{\text{TS}}) : \text{chroles}(\phi, R) = R \setminus_R \text{Role}(\phi) \wedge \text{chprotocols}(\phi, P) = P \setminus_P \text{Protocol}(\phi)$

and for all transitions $s \xrightarrow{\phi} t$ of media structures s and t

$$\begin{aligned}
\text{chroles}(\alpha_R(\phi), J_R(\text{Th}_R, (\Sigma_R, \text{Roles}))) &= J'_R(\text{Th}'_R, (\Sigma'_R, \text{Roles}')) \\
\text{chprotocols}(\alpha_P(\phi), J_P(\text{Th}_P, (\Sigma_P, \text{Protocols}))) &= J'_P(\text{Th}'_P, (\Sigma'_P, \text{Protocols}')) \\
\text{chchannels}(\alpha_C(\phi), J_C(\text{Th}_C, (\Sigma_C, \text{Channels}))) &= J'_C(\text{Th}'_C, (\Sigma'_C, \text{Channels}')) \\
\text{chchannels}(\alpha_C(\phi), J_C(\text{Th}_C, (\Sigma_C, \text{Channelsystem}))) &= J'_C(\text{Th}'_C, (\Sigma'_C, \text{Channelsystem}')) \\
\text{if } J_C(\text{Th}_C, (\Sigma_C, \text{Channelsystem})) &\xrightarrow{\alpha_C(\phi)} J'_C(\text{Th}'_C, (\Sigma'_C, \text{Channelsystem}'))
\end{aligned}$$

where

$$\begin{array}{ll}
s = C_R : L : \text{Th}_R : M_R : J_R : n_R : \text{Roles} & t = C'_R : L : \text{Th}'_R : M'_R : J'_R : n'_R : \text{Roles}' \\
C_P : L : \text{Th}_P : M_P : J_P : n_P : \text{Protocols} & C'_P : L : \text{Th}'_P : M'_P : J'_P : n'_P : \text{Protocols}' \\
C_D : L : \text{Th}_D : M_D : J_D : n_D : A & C'_D : L : \text{Th}'_D : M'_D : J'_D : n_D : A' \\
C_C : L : \text{Th}_C : M_C : J_C : n_C : \text{Channels} & C'_C : L : \text{Th}'_C : M'_C : J'_C : n'_C : \text{Channels}' \\
C_{Ch} : L : \text{Th}_C : M_C : J_C : \text{Channelsystem} & C'_{Ch} : L : \text{Th}'_C : M'_C : J'_C : \text{Channelsystem}' \\
\vdots & \vdots \\
\text{MRRoles}(C_R), \dots & \text{MRRoles}(C'_R), \dots
\end{array}$$

Let us motivate this definition. The media requirements consists of a collection of phase specific transaction languages with mappings to role, protocol and channel language.

For the transaction languages and the transitions we require some media-characteristic properties to hold: (M1) For all sentences of the language of the knowledge phase, their interpretation has to be defined. (M2) Messages of the intention phase do neither change roles nor protocols. (M3) Messages of the negotiation phase add to roles and protocols. (M4) Messages of the settlement phase reduce roles and protocols (that have been negotiated in the negotiation phase). The operations chroles and chprotocols have to be defined for all messages of negotiation and settlement phase together with $\cup_R, \cup_P, \setminus_R, \setminus_P$. Note that the phases are designed to distinguish, what is called in speech act theory the illocution of the messages [18,4]. We consider as the illocution the effect that it has on the organization. Those different sets of (transaction) languages and the characteristic properties resemble the four phases of the media model

The transitions on protocols, roles and channels have to be synchronized. Via the mapping α_x the languages are related. The operations changeprotocols , changeroles and changechannels describe how a transition changes the components of the state. Note that the organizational illocution is the same throughout media. We define that for each transition chroles and chprotocols have to be synchronized with the other views. Note that this synchronization resembles the views of the reference model that describes a medium to consist of channel system, transactions, roles, protocols and processes. Note furthermore that

the language for describing processes (the process view) is in our case the language of media structures. This language describes how organization is related to channels.

3.3 Examples

To implement a book shop on the catalog explored in 2.3, we suggest a set of transactions and assign them to the phases of the medium:

```

op send          : NameSet Domain -> Ta .      -- knowledge phase
ops supply demand : NameSet Domain -> Ta .      -- intention phase
op offer         : NameSet Ta -> Ta .           -- negotiation phase
op counteroffer  : NameSet Ta Ta -> Ta .        -- negotiation phase
ops accept reject : NameSet Ta -> Ta .          -- negotiation phase
op settle       : NameSet Ta -> Ta .           -- settlement phase.
```

We define, e.g., the role and protocol languages to abstract from the address parameter of the transactions, while the channel language needs the address as well. E.g., for the transaction **offer**

$$\begin{aligned}\alpha_R(\text{offer}(N,D)) &= \alpha_P(\text{offer}(N,D)) = \text{offer}(D) \\ \alpha_C(\text{offer}(N,D)) &= \text{offer}(N,D)\end{aligned}$$

The chrole for an **offer**(N,D) is an obligation to perform transaction D when the offer is accepted in a message **accept**(M,**offer**(N,D)), i.e.

$$\text{chrole}(\text{offer}(N,D)) = [\text{accept}(M, \text{offer}(N,D))] \text{Obl}(D)$$

The abstraction between transaction and organization may be coarser as the example of a pay message illustrates. An obligation to pay can be fulfilled either by a ECash message, by sending a check or by charging a credit card. The amount of money in the ECash and in the send-check message. I.e.,

$$\begin{aligned}\alpha_R(\text{ECash}(N1,N2),y) &= \alpha_R(\text{send-check}(N1,\text{addr}(N2),y)) = \\ \alpha_R(\text{charge}(\text{creditcard}(N1),N2,x)) &= \text{pay}(N1,N2,x)\end{aligned}$$

Note that it is still subject to specification how the organizational illocution modeled in chrole affects implicit and explicit knowledge on a medium.

To implement the book shop on the book catalog we have to model that being part of a product catalog constitutes a legally binding offer. I.e., an obligation to deliver a book is constituted from the product catalog. I.e. role Book in a bookshop is defined to be $\text{Book}(X) = \text{offer}(\text{supply}(X))$. An organization system which describes the roles of the shop to have precedence over the roles of the catalog would allow is to employ the modular specification style illustrated in 2.3.

4 Media Architecture

The formalization of the media model yields a media structure describing how to model media. The formalization of the MRM yields the media requirements

capturing basic action types in characteristic properties and the synchronization among the components defined in the media structure. Media structure and media requirements are the two basic concepts to establish a media architecture. I.e., we define a medium to

- be described following the media structure and
- fulfilling the media requirements.

The things an agent has to know to understand a medium are (1) the general logic and the way signatures, languages, models, deduction and satisfaction relation are presented and defined, (2) the media structure and (3) the media requirements description.

5 Related Work

For the design of Web- information systems, typically semi-formal graphical notations, as, e.g., the Unified Modeling Language are being employed [1,17,20]. Those descriptions lack the information necessary for artificial agents to reason about the media. The same holds for the description of services in IDL, e.g., Corba or Jini. Moreover the "binding of an agent to a place" is typically not explicitly considered in those models.

The design of agent based system focuses typically on the agent [8,9,6,14] - our focus is the medium and its structured description.

Logic and LDS have been designed to support interoperability of formalisms and to facilitate modular design of complex heterogeneous systems [2,5]. Rewriting Logic is being employed, e.g., in [3] as general framework for relating various formalisms and levels of reasoning. Note that in those models the relations between formalisms and components are not subject to reasoning and that those relations are fixed.

6 Concluding Remarks

For the design of media in application areas in Knowledge Management and EBusiness a congruence of metaphor, model and information architecture is clearly an advantage (as it is for the design of any software). It allows users and software engineers to understand, use or design the platforms. Artificial agents need a formalization of those media as well as of the behavior expected from them on a medium. For a community of *collaborating* artificial and human agents the congruence of metaphors, models and formalization is prerequisite for frictionless communication and collaboration; only when the world of artificial agents resembles the "real" world, both human and artificial agents may perform transactions based on a common "understanding" of the world. The media architecture provided in this paper can be employed (1) as information or knowledge of artificial agents and (2) as a description to implement a platform.

References

1. H. Baumeister, N. Koch, and L. Mandel. Towards a UML extension for hypermedia design. In B. Rumpe, editor, *UML 99*, Lecture Notes in Computer Science. Springer Verlag, 1999.
2. M. Cerioli and J. Meseguer. May I borrow your logic? (transporting logical structures along maps). *Theoretical Computer Science*, 28(2):311–347, 1997.
3. M. Clavel, F. Duran, S. Eker, J. Meseguer, and M.-O. Stehr. Maude as a formal meta-tool. In *The World Congress on Formal Methods in the Development of Computing Systems (FM99)*, 1999.
4. T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, D. McKay, J. McGuire, R. Pelavin, S. Shapiro, and C. Beck. Specification of the KQML agent-communication language, 1994. URL: <http://logic.stanford.edu/papers/kqml.ps>.
5. D.M. Gabbay. *Labelled Deductive Systems*, volume 1 of *Oxford Logic Guides*. Oxford University Press, 1996.
6. N.R. Jennings and M.J. Wooldridge. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.
7. U. Lechner and B. Schmid. Communities and media – towards a reconstruction of communities on media. In E. Sprague, editor, *Proc. of the 33th Int. Hawaii Conference on System Sciences (HICSS 2000)*. IEEE Press, 2000.
8. P. Maes. Modelling adaptive autonomous agents. *Artificial Life Journal*, 1(1 and 2):135–162, 1994.
9. P. Maes, A. Chavez, D. Dreilinger, and R. Guttman. A real life experiment in creating an agent marketplace. In H.S. Nwana and N. Azarmi, editors, *Software Agents and Soft Computing, Towards Enhancing Machine Intelligence*. Springer Verlag, 1996.
10. T. Maibaum. Temporal reasoning over deontic specifications. In J.-J.Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 141–202. Wiley, 1993.
11. J. Meseguer. General logics. In *Logic Colloquium*, pages 275–329. North-Holland, 1989.
12. J. Meseguer. Conditional rewriting as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.
13. J.-J.Ch. Meyer and R.J. Wieringa. Deontic logic: A concise overview. In J.-J.Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 3–16. Wiley, 1993.
14. J.P. Müller. *The Design of Intelligent Agents - A Layered Approach*. Lecture Notes in Computer Science 1177. Springer-Verlag, 1996.
15. B.F. Schmid. The concept of media. In *Workshop on Electronic Markets*, 1997.
16. B.F. Schmid. Elektronische Märkte - Merkmale, Organisation und Potentiale. In *Handbuch Electronic Commerce*. Vahlen Verlag, 1999. To appear.
17. D. Schwabe and G. Rossi. Developing hypermedia applications using OOHDM. In *Workshop on Hypermedia Development Process, Methods and Models (Hypertext'98)*, 1998.
18. P.G Searle and J.R Searle. *Speech acts*. Cambridge University Press, 1970.
19. M. Wirsing. Algebraic specification. In J.V. Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 675–788. Elsevier (North-Holland), 1990.

20. M. Wirsing, N. Koch, A. Garrido, L. Mandel, A. Helmerich, and L. Olsina. Hyper-UML: Specification and modeling of multimedia and hypermedia applications in distributed systems. In *Second Workshop on the German-Argentinian Bilateral Programme for Scientific and Technological Cooperation*, 1999.

A The Frameworks

Definition 3 (General Logic [2,11]). A 5-tuple $L = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$ is called g (general) logic. It consists of

- a category **Sign**, whose objects are called signatures,
- a functor $Sen : \mathbf{Sign} \rightarrow \mathbf{Set}$, giving the set of sentences over a given signature. The set of sentences is called Σ -language,
- a functor $Mod : \mathbf{Sign}^{op} \rightarrow \mathbf{Cat}$, giving the category of models of a given signature,
- for each $\Sigma \in |\mathbf{Sign}|$ a satisfaction relation $\models_\Sigma \subseteq |Mod(\Sigma)| \times Sen(\Sigma)$, such that for each morphism $\sigma : \Sigma \rightarrow \Sigma'$ in **Sign**, the satisfaction condition $M' \models_{\Sigma'} Sen(\sigma)(\phi) \Rightarrow Mod(\sigma)(M') \models_\Sigma \phi$ holds for each $M' \in |Mod(\Sigma')|$ and each $\phi \in Sen(\Sigma)$,
- a function $\vdash$ associating to each Σ in **Sign** a binary relation $\vdash_\Sigma \subseteq P(Sen(\Sigma)) \times Sen(\Sigma)$, called Σ -entailment, satisfying the following properties
 1. reflexivity: for any $\phi \in Sen(\Sigma)$, $\{\phi\} \vdash_\Sigma \phi$,
 2. monotonicity: if $\Gamma \vdash_\Sigma \phi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash_\Sigma \phi$,
 3. transitivity: if $\Gamma \vdash_\Sigma \phi_i$ for all $i \in I$, and $\Gamma \cup \{\phi_i | i \in I\} \vdash_\Sigma \psi$ then $\Gamma \vdash_\Sigma \psi$
 4. $\vdash$ -translation: if $\Gamma \vdash_\Sigma \phi$, then $Sen(\sigma)(\Gamma) \vdash_{\Sigma'} Sen(\sigma)(\phi)$ for any $\sigma : \Sigma \rightarrow \Sigma'$ in **Sign**,

and the following Soundness Condition is satisfied: for each $\Sigma \in |\mathbf{Sign}|$, $\Gamma \subseteq Sen(\Sigma)$, and $\phi \in Sen(\Sigma)$, $\Gamma \vdash_\Sigma \phi \Rightarrow \Gamma \models_\Sigma \phi$, where $\Gamma \models_\Sigma \phi$ iff $[(M \models_\Sigma \gamma \text{ for all } \gamma \in \Gamma \text{ implies } M \models_\Sigma \phi)]$.

A theory $Th = (\Sigma, Ax)$ is given by a signature Σ together with a set of sentences $Ax \subseteq Sen(\Sigma)$. A functor Th_0 assigns a signature Σ its category of theories $Th_0(\Sigma)$. Note that Sen and Mod can be extended to functors on theories [2].

Definition 4 (Labelled Deductive System [5]). A Labelled Deductive System is given by (A, L, R) , where A is an algebra of labels with constructors, functions and relations, L is a logical language with connectors and well-formed formulas, and R is a labelling discipline determining, how formulas, i.e., elements from L are labelled with elements from A .

$(t : \Phi)$ is called a *declarative unit*, where t is an element of A , called the label, and Φ is a set of formulas from L . A database is a declarative unit or has the form (D, F, d, U) , where D is a finite diagram of labels, d is a label, and U the set of all terms. A *diagram of labels* is a finite set of labels together with formulas $\pm R(t_1, \dots, t_n)$, such that $t_i \in D$ and R is a predicate symbol.

In this paper, we present databases as sets of declarative units with a predicate. We apply the construction of LDS several times, declarative units become formulas in a next level of LDS and abbreviate, e.g., $(l : (m : f))$ by $l : m : f$. Note that we apply this construction to databases. I.e., a set or vector of declarative unit is assigned to a vector of labels.