

Article

A Model for Information

Paul Walton

Capgemini UK, No. 1, Forge End, Woking, Surrey, GU21 6DB, UK; E-Mail: paulnicholaswalton@gmail.com; Tel.: +44-1483-764764

Received: 2 July 2014; in revised form: 27 August 2014 / Accepted: 11 September 2014 /

Published: 18 September 2014

Abstract: This paper uses an approach drawn from the ideas of computer systems modelling to produce a model for information itself. The model integrates evolutionary, static and dynamic views of information and highlights the relationship between symbolic content and the physical world. The model includes what information technology practitioners call “non-functional” attributes, which, for information, include information quality and information friction. The concepts developed in the model enable a richer understanding of Floridi’s questions “what is information?” and “the informational circle: how can information be assessed?” (which he numbers P1 and P12).

Keywords: information; philosophy; information quality; information friction; modelling

1. Introduction

In the Information Technology (IT) industry, questions about complicated systems (including variants of “what is the system?”) are addressed by developing a variety of models [1]. This paper takes the same approach to information itself and the question “what is information?” This question was asked by Floridi in [2] and has since triggered what the authors of [3] call a “cottage industry”.

Section 2 examines the relationship between information and selection. Selection processes drive improvements in efficiency and quality. Research in psychology provides a revealing example and indicates why people find information a difficult concept to understand.

Section 3 develops the model for information (MfI). It includes tools for modelling different kinds of connection which, together, describe how physical processes and information artefacts are related. Starting with direct connections as a result of physical processes, MfI shows how abstract concepts are related to the physical world. The model integrates evolutionary, dynamic and static views.

Computer systems modelling distinguishes between functional attributes of systems (the behaviour of a system) and non-functional (the qualities of a system—like performance, availability, scalability, maintainability). Within MfI, non-functional attributes correspond to information quality (IQ) and information friction (IF). IF is a measure of the resources needed for some information-related task. The discussion about IQ uses ideas about scientific measurement applied to the geometry implicit in information to define some measures of information quality. IQ is addressed in Section 4 and IF in Section 5.

In [2], Floridi articulated a number of questions about information. In Section 6, the question “what is information?” is addressed (this is Floridi’s question P1). MfI also sheds some light on some of Floridi’s other questions. Since MfI is based on the relationship between information and the physical world it also tackles the following question directly: “the informational circle: how can information be assessed? If information cannot be transcended but can only be checked against further information—if it is information all the way up and all the way down—what does this tell us about our knowledge of the world?” This question (P12) is also addressed in Section 6.

2. Information and Selection

Many authors (for example [2,4,5]) have looked at the relationship between information, selection and evolution. Examples of selection include, for example: natural selection [6], market selection (through which the “invisible hand” [7] operates), political selection through voting (in democracies) and personal selection by which people make choices.

These types of selection all have some elements in common. The form of an entity is determined by a process which is, to some extent, governed by a design pattern (using the term in the sense of [8] rather than in the (related) sense of [9]). The patterns include DNA, system architecture or organisational operating model.

Versions of entities are derived from other versions through some physical process using the pattern. Children are derived from their parents, new versions of computer systems are based on design changes to the original system and organisations develop by changing their structure, processes, roles, locations and so forth. There may be no derived entity, one, several or many for any original entity. In some cases the derived entity may be a completely new entity (“reproduction” in biological terms); in some cases, it may be the existing entity (“growth” in biological terms).

All of these processes involve interaction between an entity and its environment. Each interaction changes the properties of the entity and the environment. In some cases, the interactions of an entity impact, or rely on, the health of another entity or entities. Examples of this include a spouse, a parent and an organisation using a computer system. In these cases, a favourable outcome for the entity relies on the health of the other entity or entities.

As a result, a selection process can be considered as two feedback mechanisms. In the first one, interactions impact the properties of an entity and its ability to achieve favourable outcomes. In the second one, the interactions impact the pattern of the derived entity.

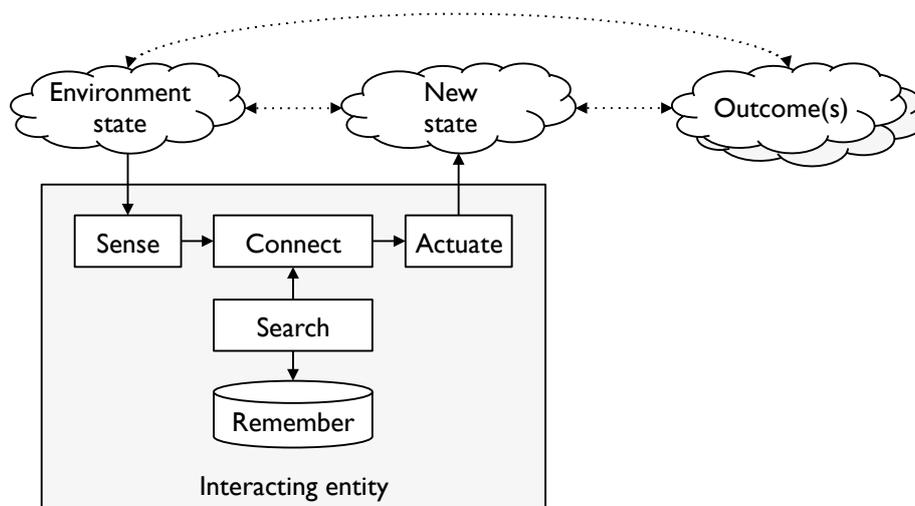
For the rest of the paper we will refine the use of the term *Interacting Entity (IE)* to refer to an entity which participates in such a selection process. Humans, animals, computer systems and organisations are all examples of IEs in this sense. To improve the favourability of outcomes, such IEs

require the following capabilities:

- to *sense* what is taking place in the environment
- to *actuate*—to cause an impact on the environment
- to *remember, search and retrieve* what has been remembered
- to *connect* the various elements together to enable the IE to link environment states to actions and potential outcomes and choose more favourable alternatives from the options.

These elements indicate the form of a generic IE model shown in the following diagram.

Figure 1. A generic Interacting Entity (IE) model.



(In [2], Floridi divided information into three categories: “information as reality”, “information about reality” and “information for reality”. If we call them types 1, 2, and 3, then the generic IE model highlights the role of types 2 and 3.)

Other things being equal, an increase in information quality will generate more favourable outcomes. However, improvements in efficiency may also generate more favourable outcomes.

An example of this is provided by research in psychology summarised by Kahneman in [10]. He describes two modes of thinking he calls “System 1” and “System 2”. As he puts it:

“System 1 operates automatically and quickly, with little or no effort and no sense of voluntary control. System 2 allocates attention to the effortful mental activities that demand it, including complex computations. The operations of System 2 are often associated with the subjective experience of agency, choice and concentration.”

“...most of what you (your System 2) think and do originates in your System 1, but System 2 takes over when things get difficult, and it normally has the last word.

The division of labour between System 1 and System 2 is highly efficient: it minimizes effort and optimizes performance. ... System 1 has biases, however, systematic errors that it is prone to make in specified circumstances.”

This has a major impact on the way in which we respond to information. Kahneman says:

“When information is scarce, which is a common occurrence, System 1 acts as a machine for jumping to conclusions.”

“System 1 is radically insensitive to both the quality and quantity of information that gives rise to impressions and intuitions.”

The distinction between Systems 1 and 2 highlights one of the impacts of selection pressures. Changes may improve efficiency or quality or both, but their reconciliation is not always elegant.

In the natural world, evolution has driven the creation of numerous ecosystems. The same is true of information—selection processes drive the creation, development and specialisation of *information ecosystems* (usually referred to as just *ecosystems* where the context is clear). As the authors state in [3]: “every separate realm of knowledge has legitimately distinguished a separate form of information.” English speakers can communicate in English with each other but not (necessarily) with non-English speakers. Mathematicians can communicate with each other in a form incomprehensible to non-mathematicians. Each ecosystem includes some types of IE and includes the conventions by which they communicate. This topic is discussed more in Section 4.

3. The Model for Information (Mfi)

In this section we use an approach based on that used in the discipline of technical architecture (see, for example, [1]). The specific purpose is to link information with the physical world so the overall model describes the following types of model and their relationship:

- Physical Connection Models (PCM) which model physical entities and their interactions;
- Extended Connection Models (ECM) which model aggregations of entities and interactions;
- Virtual Connection Models (VCM) which model more abstract connections;
- Information Artefact Models (IAM) which model Information Artefacts (IAs).

Starting with PCM, which models the physical world, these models show how elements of abstraction can be successively layered in top of the basic ideas in a way which links back to the physical world. The models are built on interactions and different types of connection between entities, so first we include some definitions which support the development of the ideas.

3.1. Definitions

Call a *slice* a contiguous subset of space-time. A slice can correspond to an entity at a point in time (or more properly within a very short interval of time), a fixed piece of space over a fixed period of time or, much more generally, an event which moves through space and time. This definition allows great flexibility in discussing information. By using slices rather than space and time separately we can apply a common approach in different cases. For example, slices are sufficiently general to support a common discussion of both nouns and verbs, the past and the future.

We also use the following to describe the physical world:

- *properties*: each slice has a set of properties and for each property there is one or more physical processes (which we call measurement) that takes the slice as input and generates the value of the property for the slice;
- *entity*: a type of slice corresponding to a substance;
- *interaction*: a physical process that relates the properties of two entities.

Interactions use different physical processes including: electromagnetism (e.g., light), other waves (e.g., sound, ripples in water), physical contact (e.g., billiard balls striking each other), chemical signals in a medium (e.g., smell), gravity and radioactivity. In all of these examples, except physical contact, interaction is *directional* in that a physical process is initiated in one entity and then impinges on another (or many more). In the other cases we can consider an interaction between x and y to consist of two separate directional interactions, one from x to y and the other from y to x . Call a directional interaction a *d-interaction*.

Any d-interaction takes place at a specific time (or over a specific time period), but the same type of interaction may take place many times. For example, there are many examples of light being emitted from one entity and absorbed by another. Call these *types* of interaction or d-interaction. D-interaction or interaction types can have the same properties as interactions or d-interactions (respectively) with the exception of space-time properties. For this reason it is useful to consider the space-time properties to be relative.

Define a *pointer* to be a tuple (l, v, S) for which:

- l is a label: $\text{lab}(c)$;
- S is a set which is the set of all possible values of the pointer: $\text{cont}(c)$;
- v is the value (a member of S or \emptyset): $\text{val}(c)$.

Pointers can be *instantiated* or not according to whether or not v isn't \emptyset .

If x is a slice, then define:

- the set of *properties of x* , $P_x = \{p_i: p_i \text{ is a pointer representing the } i\text{-th property of } x\}$;
- the *number of properties of x* , $|x| = |P_x|$;
- a *disposition δ* is an $|x|$ -tuple $(\text{val}(p_1), \dots, \text{val}(p_{|x|}))$, $p_i \in P_x$;
- the *set of all dispositions of x* , or *disposition space of x* , Δ_x .

When the properties of an entity meet certain conditions, then an interaction will be triggered. If α is a d-interaction type from entity x to y , then not all of the properties of x and y necessarily affect or are affected by α . Define:

- $P_{x\alpha}$ the set of properties which can trigger α (“o” stands for output);
- $\Delta_{x\alpha}$ the corresponding disposition space (the projection of Δ_x onto $P_{x\alpha}$);
- $P_{y\alpha}$ the set of properties altered by α (“i” stands for input);
- $\Delta_{y\alpha}$ the corresponding disposition space (the projection of Δ_y onto $P_{y\alpha}$).

Then for any x and y , α has the following two associated functions:

- the output function, $\mathbf{o}\alpha: \Delta_{x\alpha} \times T \rightarrow \Delta_{x\alpha}$ which maps any disposition of x before α is output to the disposition afterwards;
- the input function, $\mathbf{i}\alpha: \Delta_{y\alpha} \times T \rightarrow \Delta_{y\alpha}$ which maps any disposition of y before α is input to the disposition afterwards.

We can write:

$$\mathbf{o}\alpha: (\delta_x, t) \rightarrow \delta_{x\alpha} \text{ for } \delta_x \text{ in the domain of } \mathbf{o}\alpha;$$

$$\mathbf{i}\alpha: (\delta_y, t) \rightarrow \delta_{y\alpha} \text{ for } \delta_y \text{ in the domain of } \mathbf{i}\alpha.$$

We can define a *d-interaction function* $\mathbf{t}\alpha$ as follows:

$$\mathbf{t}\alpha: \Delta_{x\circ\alpha} \times \Delta_{y\circ\alpha} \times T \rightarrow \Delta_{y\circ\alpha} \text{ and } \mathbf{t}\alpha(\delta_x, \delta_y, t) = \delta_{y\circ\alpha}.$$

α is a *transmission* if:

$$\mathbf{t}\alpha(\delta_x, \delta_y, t) = \mathbf{t}\alpha(\delta_x, \delta'_y, t) \text{ for all } \delta_y, \delta'_y \in \Delta_y \text{ and for all } t \in T.$$

The effect of a transmission does not rely on the disposition of the destination entity or time, only on the nature of the *d-interaction* itself. In this case we can write

$$\mathbf{t}\alpha: \Delta_{x\circ\alpha} \rightarrow \Delta_{y\circ\alpha} \text{ and } \mathbf{t}\alpha(\delta_x) = \delta_{y\circ\alpha}.$$

Examples of transmissions include writing to computer memory and transmissions in the more usual sense (e.g., transmission of electronic messages).

Suppose that $\alpha_1, \dots, \alpha_{n-1}$ is a series of *d-interaction* types connecting entities x_1, \dots, x_n . They have the same effect as a single *d-interaction* if the range of one *d-interaction* input function is contained in or equal to the domain of the next output function. The effect on x_1 and x_n is the same as if there was a *d-interaction type*, β , connecting x_1 and x_n . In this case, though the *transmission function* $\mathbf{t}\beta$ has the following form:

$$\mathbf{t}\beta: \Delta_{x_1\circ\alpha_1} \times \dots \times \Delta_{x_{i-1}\circ\alpha_{i-1}} \times \dots \times \Delta_{x_n\circ\alpha_n} \rightarrow \Delta_{x_n\circ\alpha_n} \text{ and } \mathbf{t}\beta(\delta_1, \dots, \delta_i, \dots, \delta_n) = \delta_{n\circ\alpha_n}.$$

As Floridi discusses in [2], abstraction is concerned with leaving out detail. We use the term “pattern” for this. In this case, the term is linked to the usage in the computer systems architecture world [8] rather than the sense in which is it used by Burgin [11], for example.

To make this more precise and to describe structures and connections more generally, we use an approach based on graph theory. This approach is consistent with the approach (using static schemas) used by Burgin in [11] but allows a more natural use of some graph theory-based ideas.

An *unrooted linnet* (where “linnet” stands for “linked net”) C is a tuple (V, E, N) which satisfies the following conditions:

- V is a finite set of *vertices*;
- E is a finite set of *connectors*;
- V and E are disjoint;
- N is the set of *connections* of D where $N \subseteq \{(a, e, b): a, b \in V \cup E, e \in E\}$;
- if $e \in E$ and $(a, e, b), (a', e, b') \in N$, then $a = a'$ and $b = b'$.

For compatibility with graph theory, if (v, e, w) is a connection with $v, w \in V, e \in E$ then e is called an *edge* (so an edge joins two vertices). (Since (v, e, w) is an ordered set, more properly it should be a *directed edge*.) The vertices, connectors and connections of C are known as $V(C), E(C)$ and $N(C)$ respectively.

Define the *proximity* of a vertex to be 0. If (e_1, e, e_2) is a connection, then the proximity of e , $\text{prox}(e)$, is defined as follows: if (e_1, e, e_2) is a connection then $\text{prox}(e) = \min(\text{prox}(e_1), \text{prox}(e_2)) + 1$ where \min is undefined if either $\text{prox}(e_1)$ or $\text{prox}(e_2)$ is undefined. A connector is *rooted* if its proximity is defined. A *linnet* is an unrooted linnet in which all connectors are rooted. So in a linnet all connectors connect back to vertices.

If C and D are linnets, then (by analogy with graphs—see [12]) C is a *sub-linnet* of D and D is a *super-linnet* of C if:

- $V(C) \subseteq V(D)$;
- $E(C) \subseteq E(D)$;
- $N(C) \subseteq N(D)$.

In the examples we will be interested in below, the sets $V(C)$ and $E(C)$ will contain pointers—call the linnet in this case *pointed*, otherwise *unpointed*. From here on only pointed linnets will be considered and the term “linnet” will apply to pointed linnets unless otherwise qualified.

When all connections in C are edges, then C is a directed graph. (More precisely and using the terminology of [12], C is a directed multi-graph since more than one edge can connect two vertices.)

A *dynamic linnet* DC is a set of linnets $\{C_i: t_i \in T\}$ for which T is a set of times. Call

$$V(DC) = \cup V(C_i) \text{ and } E(DC) = \cup E(C_i)$$

the vertices and connectors of DC . A dynamic directed graph (DDG) is a dynamic linnet for which each C_i is a directed graph. (Dynamic graphs are discussed in [13].)

If C is a (pointed) linnet, then C is:

- *instantiated* if all the pointers in $V(C)$ and $E(C)$ are instantiated;
- *uninstantiated* if all the pointers in $V(C)$ and $E(C)$ are uninstantiated;
- *partially instantiated* if it is not instantiated.

Call a partially instantiated linnet a *pattern* (by analogy with the computer systems architecture term). Patterns are closely related to queries and searching. For example, any query in the Structured Query Language, SQL [14], has a corresponding pattern.

Burgin provides a thorough discussion of structures in [11]. He shows that his definition of structure, defined in terms of static schemas, is sufficiently rich to model structures associated with information. If vertices in linnets are associated with object variables in static schemas and connectors are associated with connection variables, then linnets have the same potential.

3.2. Physical Connection Models

A Physical Connection Model (PCM) is a dynamic linnet which represents entities and the d-interactions between them. The rationale for using pointers in linnets is that it allows us to separate two different purposes of each vertex or connector. In each case the pointer label allows consumers of the model (e.g., readers of this paper) to identify the different components. The pointer value represents the value accessible by other elements within the model itself. In other words, the label and value are accessed and used by (potentially) different entities and processes inhabiting (potentially) different ecosystems.

We can define the Physical Connection Linnet (PCL), C_k , for entities $\{x_i\}$ and their d-interactions $\{e_j\}$ at time t_k in the following way:

- let $v(x_{ik})$ be the pointer with label x_i and value $\delta_{x_{ik}}$, where $\delta_{x_{ik}}$ is the disposition of x_i at time t_k ;
- let $e(e_{jk})$ be the pointer with label e_j and value $\delta_{e_{jk}}$, where $\delta_{e_{jk}}$ is the disposition of e_j at time t_k ;
- let $V(C_k) = \cup_i \{v(x_{ik})\}$ and $E(C_k) = \cup_j \{e(e_{jk})\}$;

- $N(C_k) = \{(v, e, w) : v, w \in V(C_k), e \in E(C_k), \text{ the } d\text{-interaction represented by } e \text{ connects the entities represented by } v \text{ and } w\}$;
- $C_k = (V(C_k), E(C_k), N(C_k))$.

A set of such PCLs for times $\{t_k\}$ is a dynamic linnet called a *Physical Connection Model (PCM)*. Any set of entities, the d-interactions that connect them and a set of times induce a unique PCM.

3.3. Extended Connection Models

A PCM models individual d-interactions between entities, but in many cases, there are more complicated forms of connection. Suppose that:

- $X = \{x_j : 1 \leq j \leq n\}$ is a set of entities;
- $Y = \{x_j : 1 \leq j \leq m\}$ for some $m < n$;
- A is a set of interactions between elements of X ;
- $A|Y = \{\alpha \in A : \alpha \text{ is an interaction between elements of } Y\}$.

X and A induce a linnet as well as Y and $A|Y$. However, we can also form a linnet from $X \setminus Y$ and $A \setminus (A|Y)$ by contraction. If:

- C is a linnet;
- e is an edge of C corresponding to the connection (v_1, e, v_2) ;
- $v \notin V(C) \cup E(C)$

then the *contracted* linnet $C \setminus e$ is defined as follows:

- $V(C \setminus e) = (V(C) \cup \{v\}) \setminus \{v_1, v_2\}$;
- $E(C \setminus e) = E(C) \setminus \{e\}$;
- $N(C \setminus e) = N(C) \setminus \{n : n \in N(C), e \in n\}$;
- any occurrence of v_1 or v_2 in C is replaced with v in $C \setminus e$;
- any occurrence of e in C is replaced with v in $C \setminus e$.

Since contraction can only decrease the proximity of a connector, this is a linnet.

Generalising the terminology in [12], $C \setminus e$ is known as an *elementary contraction* of C . A linnet D is *contractible* to a linnet C , or is a *contraction* of a linnet C , if C can be obtained from D by a sequence of elementary contractions. By extension we can define $C \setminus F$ for any subset F of edges of $E(C)$.

So, to return to the example above, if C is the linnet induced by X and A , F is the set of edges of C corresponding to $A|Y$. Then $C \setminus F$ is a linnet. So we can treat the set of entities Y as a single aggregated entity for which

$$\Delta_Y = \Delta_{x1} \times \dots \times \Delta_{xm}.$$

More generally, we can support the definition of arbitrary nesting of entities.

If C_k is a PCL for some time t_k , and $EC = \{C_k\}$ is the corresponding PCM, then we can create an *Extended Connection Model (ECM)* by contraction. If $E \subseteq E(EC)$ define $EC_k = C_k \setminus (E \cap E(C_k))$; EC_k is an *Extended Connection Linnet (ECL)*. $\{EC_k\}$ is an ECM which we can call $EC \setminus E$. With this definition, the individual components of an IE (see Figure 1) are ECMs and the IE as whole is an ECM.

3.4. Connections and Relations

PCMs and ECMs model entities and their interactions but there are other types of connection we want to consider. Consider first the properties of slices. Since properties are something that can be measured there is a physical process which takes the slice as input and creates an *Information Artefact* (IA) as output which represents the property.

This appears to be the start of an infinite regress. A property of one entity can be explained by a process which creates a specific property in another entity. However, the IA belongs in an ecosystem (all IAs and processes that create them belong in an ecosystem) so we will investigate ecosystems and symbols before returning to this question.

In the natural world, evolution has driven the creation of numerous ecosystems. In each case there are a set of conventions which apply to relevant IEs and what they are communicating. The scope of these conventions includes:

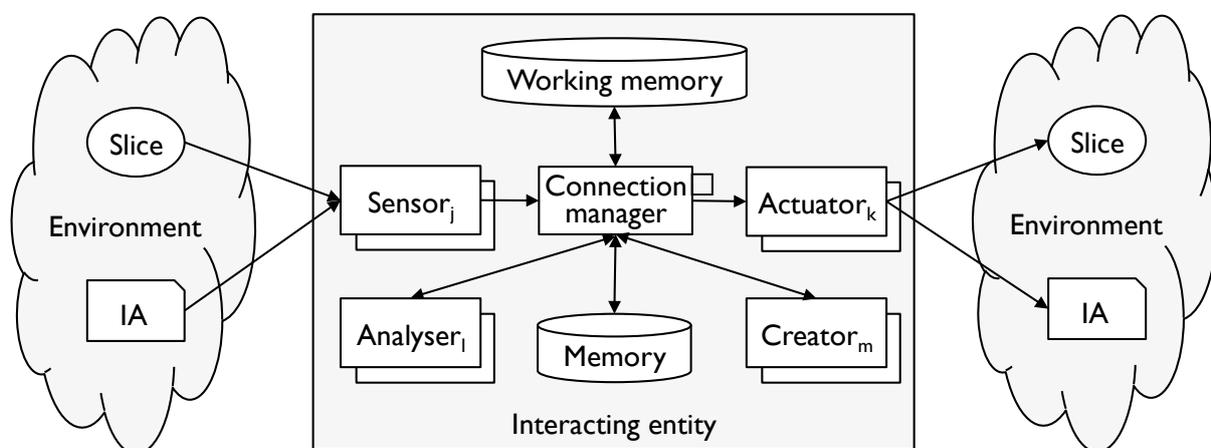
- the symbols which are used;
- the structure of IAs and the rules that apply to them;
- the ways in which concepts are connected;
- the syntax of IAs and how they are created;
- the channels which are used to interact (and the associated sensor and actuator patterns).

Each of these elements co-evolves with the others.

Call such a set of conventions a modelling tool. Languages and Mathematics are modelling tools, for example. There is a close relationship between information ecosystems and modelling tools. We can define an information ecosystem to be the set of IEs, IAs and channels that support a particular modelling tool. (Of course, a particular ecosystem may be a subset of other ecosystems and therefore support other modelling tools.)

Use of a modelling tool requires special processing from IEs, so we can redraw the elements of the generic IE model as in Figure 2.

Figure 2. The generic IE and modelling tools.



Working memory holds transitory information as it is worked on. Analysers take the output of a sensor and convert it into a standard form appropriate to the modelling tool. Creators do the opposite

for an actuator. In particular, they detect and aggregate symbols using the rules of the modelling tool.

We can define these concepts in terms of the discussion of the paper to date, starting with symbols. Suppose that:

- E is a set of IEs in an ecosystem corresponding to a modelling tool M;
- A is a set of IAs in the ecosystem;
- AP is a pattern for each instance of the M analysers for all $e \in E$ (the existence of a modelling tool and ecosystem guarantees the existence of such a pattern).

Then:

$\forall e \in E, a \in A, \exists \delta_a \in \Delta_a$ such that P_e maps each δ_a to the same disposition δ_e and each δ_e represents each other δ_e with respect to all M interactions.

Any such δ_a is then a *symbol instance* and the set $\{\delta_a\}$ is a *symbol type*.

What this means is that the analysers map all “symbol” dispositions representing the symbol to the same thing and they are interchangeable in ecosystem interactions. Even when the formats are completely different, the modelling tool patterns in M ensure that they are convertible between each other. MTs also have patterns for structuring symbols in IAs and IEs so that there are MT conventions for their relative structure (e.g., sequence). Any set of symbol instances which meet such structuring requirements we call a *content instance*. Similarly we can define a *content type*. Call any disposition which is contained in a content type *symbolic*. Not all dispositions are symbolic.

With these definitions we can convert symbolic dispositions to symbolic content and, in later elements of MfI, refer to content within the context of a modelling tool in a way which is consistent with the earlier models about properties and entities.

So, in MfI, a *modelling tool M* is a tuple (I, S, Cp, A, Ch) defined by the following:

- a set of IE patterns, $I = IE(M)$ (each including patterns for sensing, actuation, analysing, creating and connecting);
- a set of symbol types, $S = ST(M)$;
- a set of content patterns, based on $ST(M)$, $Cp = CT(M)$;
- a set of IA patterns, based on $ST(M)$, $A = IA(M)$;
- a set of channel patterns, $Ch = CH(M)$;

for which:

- each sensor instance consistent with a pattern in $IE(M)$ consumes IAs which are consistent with a pattern in $IA(M)$ over a channel consistent with a pattern in $CH(M)$;
- each actuator instance consistent with a pattern in $IE(M)$ produces IAs which are consistent with a pattern in $IA(M)$ over a channel consistent with a pattern in $CH(M)$;
- each IA consistent with a pattern in $IA(M)$ is consumed or produced by all IEs consistent with at least one pattern in $IE(M)$;
- each IA consistent with a pattern in $IA(M)$ contains content consistent with at least one pattern in $CT(M)$;
- each channel consistent with a pattern in $CH(M)$ is used by at least one IE consistent with a pattern in $IE(M)$.

In other words, the pieces of M fit together.

From the definitions, it seems that the definitions are mutually dependent since symbols rely on modelling tools and vice versa. However, the reality is that they co-evolve over time.

If M is a modelling tool, then there is an associated *information ecosystem* $Eco(M) = (E, A, C)$ for which:

- $E = \{e: e \text{ is an instance of some pattern in } IE(M)\};$
- $A = \{a: a \text{ is an instance of some pattern in } IA(M)\};$
- $C = \{c: c \text{ is an instance of some pattern in } Chan(M)\}.$

Conversely, for any information ecosystem E, there is a corresponding modelling tool $M = MT(E)$ for $E = Eco(M)$.

With these pieces in place we can now continue the discussion started earlier about relations and types of connection. Any property of a slice or slices or relation between any number of slices corresponds to a physical process as shown in the following Table 1.

Table 1. Types of connection as physical processes.

#	Type of connection	Equivalent as physical process
1	Physical process instance	Physical process instance (as in PCMs or ECMs)
2	m slices satisfy relation	There is an ecosystem process, which is an instance of a modelling tool pattern, which takes m slices as input and generates a content instance which is part of the content type “true” as output when the relation is satisfied
3	m slices have a property	There is an ecosystem process, which is an instance of a modelling tool pattern, which takes m slices as input and generates a content instance which is part of the ecosystem content type as output representing the value of the property

For any connection of types two and three in the table, we can say that the connection *supports* entities and processes of particular types if the corresponding physical process or process type (in the right hand column of the table) requires the corresponding ecosystem patterns.

3.5. Virtual Connection Models

Virtual Connection Models (VCMs) differ from PCMs and ECMs in that the types of connection shown in Table 1 are included and ecosystem content, as described above, is also represented. These ideas are applied to the generic IE model. The term “virtual” is used because, with the exception of connections representing physical process instances, the additional types of connection represent “what ifs”. They define what would happen if a particular process type was applied to particular slices.

Let C be a content type corresponding to some connections defined in the manner of Table 1. We can define the Virtual Connection Linnet (VCL), C_k , for entities $\{x_i\}$ and their d-interactions $\{e_j\}$ and connections C at time t_k in the following way.

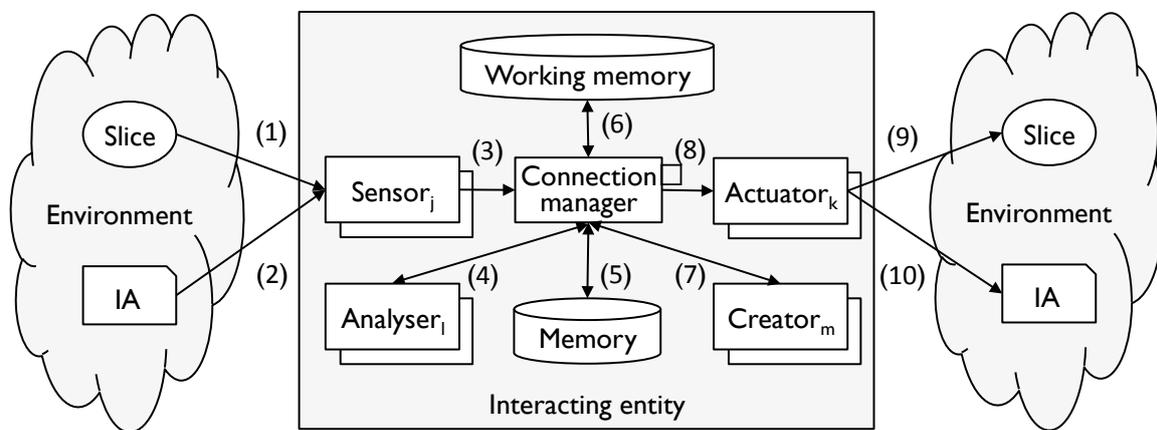
- let $v(x_{ik})$ be the pointer with label x_i and value $\delta_{x_{ik}}$, where $\delta_{x_{ik}}$ is the disposition of x_i at time t_k ;
- let $e(e_{jk})$ be the pointer with label e_j and value $\delta_{e_{jk}}$, where $\delta_{e_{jk}}$ is the disposition of e_j at time t_k ;
- let $c(c_{lk})$ be a pointer with label $lab(c(c_{lk})) \in C$ and value representing the relevant ecosystem process;

- let $V(C_k) = \cup_i \{v(x_i)\}$, $E_1(C_k) = \cup_j \{e(e_{jk})\}$, $E_2(C_k) = \cup_l \{c(c_{lk})\}$ and $E(C_k) = E_1(C_k) \cup E_2(C_k)$;
- $N_1(C_k) = \{(v, e, w) : v, w \in V(C_k), e \in E_1(C_k), \text{ the } d\text{-interaction represented by } e \text{ connects the entities represented by } v \text{ and } w\}$;
- $N_2(C_k) = \{(x, e, y) : x, y \in V(C_k) \cup E(C_k), e \in E_2(C_k), \text{ the process represented by } e \text{ supports } x \text{ and } y\}$;
- $N(C_k) = N_1(C_k) \cup N_2(C_k)$;
- $C_k = (V(C_k), E(C_k), N(C_k))$.

A set of such VCLs for a set of times $\{t_k\}$ is a dynamic linnet called a *Virtual Connection Model (VCM)*. Note that any PCM or ECM is also a VCM.

Figure 3 shows the relationship between a generic IE and its environment with the interfaces between elements numbered.

Figure 3. A generic IE in the context of Virtual Connection Models (VCM).



The different stages are as follows:

- (1) *Environment input*: an input interaction with the environment;
- (2) *IA input*: an input interaction with the environment in which the source of the input is an IA;
- (3) *Converted input*: the input converted into a form which can be manipulated by the IE;
- (4) *MT-converted input*: the input converted into a form consistent with the relevant modelling tool (if any);
- (5) *Memories*: material collected from memory;
- (6) *Attention*: the short-term picture created by the IE in working memory;
- (7) *MT-converted output*: output converted into a form consistent with the rules of a modelling tool;
- (8) *Converted output, environment output, IA output*: the reverse of the corresponding input.

In a VCM we can model memory by defining the inputs and outputs in terms of linnets. In principle (and subject to physical limitations) any linnet can be stored, but it is important to note that there are two different types. The output from an analyser is symbolic, whereas environment input is not.

Storage of these different types corresponds to different types of human and computer memory. Humans have semantic memory and episodic memory [15]. Computer databases include symbolic records and also binary large objects (things like media files). For IEs in general we use the terms *symbolic memory* and *event memory* to distinguish between them.

Retrieval of memories faces a combinatorial problem so selection pressures demand a way of specifying a manageable portion. In computing, languages like SQL [14] have been defined for that purpose. In MfI, queries are based on patterns which are linnets in which there is the potential for pointers to be uninstantiated. The response to a query is an aggregated linnet which contains copies of linnets in memory that match the pattern. The scope of the query pattern has a large impact on the results. A larger linnet pattern (in the sense of super-linnet) will constrain the search more.

3.6. Information Artefact Models

IAs are physical entities and their behaviour as physical entities can be incorporated in ECMs. However, they are also the repositories of ecosystem content. Information Artefacts Models (IAMs) will consider this aspect. Previous sections have shown how content is related to physical properties, so IAMs support just content connections. Each modelling tool, M , has a set of IA patterns, $IA(M)$ —each of which is a linnet—which defines the permissible structures for M in terms of the symbol types of M , $ST(M)$.

Symbols can correspond to characters (as in written language), gestures (including some human or animal gestures or, in a more complex example, sign language), images (as in computer icons) or, more generally, any slice. So in the light of the discussions above we need to be precise about content pointers and how we represent them.

Define:

- $SS(M)$ to be the set of ordered sequences of elements of $ST(M)$;
- $CS(M)$ to be the subset of $SS(M)$ containing permissible modelling tool content.

For a content pointer c , we know that:

- $lab(c) \in CS(N)$ (where N is the modelling tool for consumers of the model);
- $val(c) \in CS(M)$.

(Note that labels are in a format which is accessible to consumers of the model (e.g., readers of this document). However, values are not necessarily in such a format. When we use them in text or diagrams we use an element of $CS(N)$ to represent an element of $CS(M)$ even where $CS(N)$ and $CS(M)$ are different. This is reasonable so long as we are aware of the requirements of the underlying model.)

Suppose that:

- $ap \in IA(M)$ for a modelling tool M ;
- $UV(ap)$ is the set of uninstantiated pointers in ap .

Then:

- *instantiating* $c \in UV(ap)$ means that $val(c)$ is given the value vc for some $vc \in CS(M)$;
- an *Information Artefact Linnet* (IAL) conforming to ap is obtained by instantiating all elements of $UV(ap)$.

All IALs are obtained in this way. A set of such IALs for a set of times $\{t_k\}$ is a dynamic linnet called an *Information Artefact Model* (IAM).

The discussion above relates IAs to a single modelling tool whereas in reality different modelling tools can be used. For example, a document may contain English text and calculus, or text and images.

So we need to define a *Composite Information Artefact (CIA)*, A_0 , which is an IA conforming to modelling tool MT_0 , which is instantiated (in part, at least) with pointers whose values are IAs, A_i for $i > 0$, conforming to modelling tool MT_j . In turn the A_i may be instantiated with IAs conforming to other modelling tools and so forth.

4. Information Quality (IQ)

This section discusses Information Quality (IQ) in the context of Mfl. We are all familiar with reasons why the quality of information degrades. For example:

- the human memory is fallible;
- digital media wear out and fail (as anyone who has suffered a computer disc head crash can attest);
- transmission over a network can be unreliable.

It pays to be careful about IQ. Kahneman [10] presents a clear warning about the difficulties that people have with IQ and there are entire areas of human experience which rely on those difficulties. Magic provides a good example; in [16], the authors show how the tricks involved in magic are linked to neuroscience.

IQ is in the same position as information as a whole—there is no clear agreement as to what it is. Burgin [11] provides a thorough review. So we need to be clear what we mean by information quality (IQ) and how it can be measured. If x is an IA we can ask the following questions:

- how well does x represent what it is intended to represent?
- how useful is x in what an IE needs to achieve?

Call the answer to the first of these *intrinsic IQ* and the second *action IQ*. However, to investigate both of these we need to look at *interpretation*—how information relates to the world.

4.1. Interpretation

Section 3 above shows that the relationship of information with the physical world takes place in IEs and, more specifically, in the connections made in working memory. However, what is the nature of those connections? And what is being connected?

We can understand more about information by examining the relationship of information with the physical world. One possibility is that information describes one or more slices—the subject of the information. So, for example, “grass is green” describes grass. More generally, in the traditional view of grammar, the predicate provides a property that the subject is characterised by.

But a model for information should support all kinds of information. So also consider, for example, scientific laws encoded as mathematical equations and how they apply to the world. In this case, what is being described: the right hand side of the equation or the left? Clearly in this case the equation is stating that two things, described in different ways, are the same.

The same idea applies, in a sense, to language. “Some grass is green” and “some green things are grass” are logically equivalent and provide the same information, but have different subjects. This means that the direction is a function of language to focus attention on the subject, not fundamental to

the inherent information. When the direction matters we can refer to the *subject* and *qualifier*.

Whether or not the direction is relevant depends on the purpose of the information. Consider the following examples:

- the whodunit question: what is the “something” (the subject) that some information describes?
- the completeness question: what other information is there about the subject?

Both of these have a subject. However, in all cases, information encodes a comparison (of some kind—not always equality) between two set of slices characterised by their properties. So we can say that the purpose of type 2 information is to make statements about the set theoretic relationships between slices which have different combinations of properties.

For example, consider the following sentence: $C = \text{“John from Rome wore a coat”}$. Suppose then, that:

- S_c = slices corresponding to people wearing coats;
- S_j = slices corresponding to people called John;
- S_r = slices corresponding to people from Rome.

Then C corresponds to $S_c \cap (S_j \cap S_r) \neq \emptyset$.

So, how does this relate to the description of MfI above? To understand this question better we need to look more at modelling tools. Content in modelling tools is of the following types:

- *atom*: the fundamental indivisible level of content in the modelling tool (note that this definition is different from the definition of “atom” in [17]) ;
- *chunk*: groups of atoms;
- *assertion*: the smallest piece of content that is a piece of information—again composed of two chunks in a manner that conforms to an assertion pattern;
- *passage*: a related sequence of assertions.

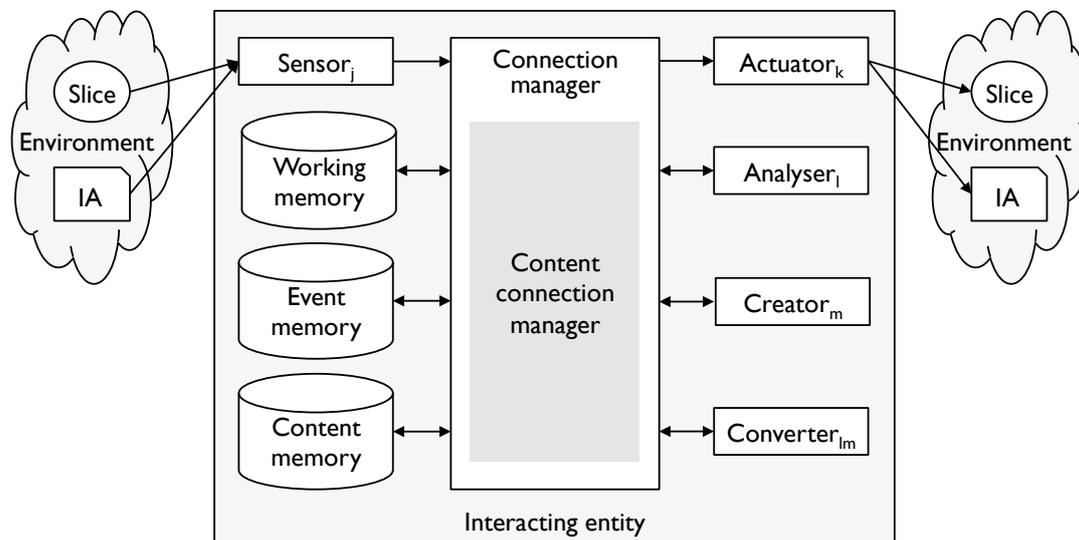
Table 2 provides some examples and shows how the different levels are interpreted. Note that not all pieces of valid language are included (for example, questions).

Table 2. Types of modelling tool content.

Component	Examples	Interpretation
Atom	Language: a word Mathematics: a variable Programming: a variable	A set of slices
Chunk	Language: a phrase Mathematics: a function with variables Programming: an expression	A set of slices that further constrains the slices corresponding to individual atoms
Assertion	Language: a sentence Mathematics: an equation Programming: an assignment statement	A set theoretic relationship between two sets of differently defined slices
Passage	Language: a paragraph (of assertions about the same things) Mathematics: simultaneous equations Programming: a function or subroutine	Multiple set theoretic relationships between sets of differently defined slices

To make this more precise let us first consider the relationship between interpretation and content. The following diagram (Figure 4) shows the generic IE model taking this into account.

Figure 4. The generic IE and content.



The diagram shows three kinds of memory. Working memory is the same as before, but persistent memory has been split into event memory and content memory. The diagram also recognises that IEs work with more than one modelling tool and may convert between them. This might include, for example:

- humans converting between French and English;
- computer programs converting between messages using a banking protocol and their database structure.

When we are dealing with content we can define variants of the transmission function since we can use content types instead of dispositions. Suppose that M and M' are modelling tools, x and y are entities and α a transmission type. Let us make the following definitions:

- Γ_M is the set of content in M ;
- Γ_{cM} is the set of chunk content in M ;
- Γ_{aM} is the set of assertion content in M ;
- Γ_{pM} is the set of passage content in M ;
- Ξ is the set of set theoretic relationships (including $\subseteq, \supseteq, \subset, \supset, \cap = \emptyset, \cap \neq \emptyset$);
- Σ is the set of all slices and $\Sigma' \in \Sigma$;
- T is a set of discrete times.

We can characterise functions corresponding to different parts of the model in Figure 4 as follows:

- *analysis* (M) $dcM\alpha: \Delta_{x\alpha} x T \rightarrow \Gamma_M$ (from disposition to content with modelling tool M);
- *creation* (M) $cdM\alpha: \Gamma_M \times T \rightarrow \Delta_{y\alpha}$ (from content to disposition with modelling tool M);
- *conversion* (M, M) $ccM\alpha: \Gamma_M \times T \rightarrow \Gamma_M$ (from content to content within modelling tool M);
- *conversion* (M, M') $ccMM'\alpha: \Gamma_M \times T \rightarrow \Gamma_{M'}$ (from content to content, converting from M to M').

Call these *content transmissions* and *content functions*. We can use the same ideas (extended to slices) to define the following types of function:

- *chunk interpretation (M)* $ciM: \Gamma_{cM} \times T \rightarrow 2^{\Sigma}$ (the set of slices which is the interpretation of a chunk with modelling tool M at a specific time);
- *assertion interpretation (M)* $aiM: \Gamma_{aM} \times T \rightarrow 2^{\Sigma} \times \Xi \times 2^{\Sigma}$ (the sets of slices and relation which is the interpretation of an assertion with modelling tool M at a specific time).
- *passage interpretation (M)* $piM: \Gamma_{pM} \times T \rightarrow 2^{\Sigma} \times \Xi \times 2^{\Sigma}$ (the sets of slices and relation which is the interpretation of a passage with modelling tool M at a specific time).

For any assertion interpretation there are two induced chunk interpretations. If i is an assertion interpretation then for any $\lambda \in \Gamma_{aM}$ and for some $t \in T$ there are sets $S_{i\lambda t}$ and $U_{i\lambda t}$ such that:

- $S_{i\lambda t} = \{s \in \Sigma' : i(\lambda, t) = (s, r, u), \text{ for some } u \in \Sigma\}$;
- $U_{i\lambda t} = \{u \in \Sigma' : i(\lambda, t) = (s, r, u) \text{ for some } s \in \Sigma\}$.

(Note that it is not the case that $i(\lambda, t) = (s, r, u) \forall (s, u) \in S_{i\lambda} \times U_{i\lambda}$.)

These definitions allow us to look at interpretation by IEs. Interpretation by an IE depends on the nature of its connections in memory and the degree to which they are exercised. Depending on the type of connection developed in working memory, a piece of chunk content will be connected to other content and memories in event memory. The net result will be the capability to match (or not) any given slice corresponding to a chunk interpretation. If two pieces of chunk content are combined in an assertion, the same capability will correspond to an assertion interpretation. Each connection constrains the set of slices that may match so the more extensive the connection the greater the constraint.

The amount of “effort” exerted by an IE is linked to the extent to which it exercises the connection and determines the constraints. Kahneman’s Systems 1 and 2 (see [10]) provide an example.

The matching is, like the connections above of which it is an example, a “what if”—it is not necessarily validated. Any interpretation produced by an IE can, in principle, be tested. (Popper [18] showed the significance of hypotheses and refutation in the scientific method and science is an example of the large scale ecosystem interpretation of information.)

4.2. Geometry and Information

The discussion of interpretation forms the foundation for a discussion of information quality. Comparisons of pieces of information with respect to some quantity lead naturally to the following questions:

- When are two pieces of information the same with respect to the quantity?
- When does one piece of information have more of the quantity than another?

In [17], the authors investigated these questions for the information algebras they define. Below we consider the questions as they apply to Mfl.

The pointers defining the domain and range of a content function α can be either continuous (normally real numbers) or discrete and either single dimensional or multi-dimensional. The nature of the pointers determines the nature of the order (if any) that applies. In some cases a total order applies—in some cases, partial. The following Table 3 provides some examples.

Table 3. Characteristics of pointers.

#	Example	Disposition pointer(s)	Content pointer(s)	Order
1	Temperature	Continuous, single dimensional	Discrete, single dimensional	Total
2	Length of the mercury in a tube	Continuous, single dimensional	Discrete, single dimensional	Total
3	A number as a finite string of digits	Discrete, single dimensional	Discrete, single dimensional	Total
4	Personal assessment of temperature into one of {cool, cold, warm, hot}	Discrete, single dimensional	Discrete, single dimensional	Partial (there may be an overlap between “cool” and “cold” for example)
5	Shape	Continuous, multi-dimensional	Discrete, multi-dimensional	Partial (induced by the total order for each dimension)
6	Species (e.g., lion, cheetah, leopard)	Discrete, single dimensional	Discrete, single dimensional	No order

Note that content, by its nature, is always discrete (even if it represents a number, it is only to a finite number of decimal places). In some cases (for example, number 4 in the table) the order is a convention of the ecosystem.

In some cases properties are preserved, in other cases not. Define *tonic* to be either monotonic or antitonic if the domain and range have the same order type (total or partial) and *p-tonic* where the domain is total and the range is partial. A function is *discriminating* if either of the following holds:

- The range is discrete;
- The range is continuous and contains two or more disjoint subsets.

The following Table 4 shows some examples for transmission functions.

Table 4. Properties of transmission functions for various transmissions.

Example	Domain	Range	Transmission function	Metric
Reading a mercury thermometer	Continuous, total order	Continuous, total order	Tonic	Yes
Reading a digital thermometer	Continuous, total order	Discrete, total order	Tonic	Yes
Personal assessment of temperature into one of {cold, cool, warm, hot}	Continuous, total order	Discrete, partial order	P-tonic, discriminating	No
Recognition of cat species into {lion, leopard, cheetah}	Continuous, multi-dimensional, no order	Discrete	Discriminating	No

4.3. Measurement

Science has some well-established concepts for measuring the quality of scientific information. This section examines whether we can generalise these to support the wider concepts of information quality. Three key measures are accuracy, precision and resolution which can be defined as follows:

- Accuracy is the degree to which a measurement is close to the true value;
- Precision defines the extent to which repeated measurements give the same result;
- Resolution defines the size of change in a property which produces a change in the measurement.

If we want to apply these concepts to MfI we need to be clear about the limitations. The measures are not usable for information more generally in the same form. They require a metric which the previous discussion demonstrates that we are not guaranteed. The “true value” is not something which is obvious from MfI. In each case there is a clear direction of representation in the content—the subject is the slice being measured and the qualifier is the measurement.

The discussion about geometry above indicates some of the limitations that apply to transmissions but similar ideas also apply to interpretation functions. The sections below look at chunks first. All of the discussion applies to an IE and a modelling tool M.

Suppose that c, c' are chunk interpretation functions, $t \in T$ and $\lambda \in \Gamma_{cM}$. Define:

- $c <_{\lambda t} c'$ if $c(\lambda, t) \subset c'(\lambda, t)$;
- $c \leq_{\lambda t} c'$ if $c(\lambda, t) \subseteq c'(\lambda, t)$;
- $c =_{\lambda t} c'$ if $c(\lambda, t) = c'(\lambda, t)$.

Then $<_{\lambda t}$ is a partial order induced by set inclusion.

The concept of “true value” is not always clear in real life. Some art forms (e.g., the “whodunit”) are based on the difficulties associated with it. Even for fundamental examples in science the nature of “true value” and how it is measured changes (think, for example, of the changes in definition of fundamental units like the metre). This shows that “true value” is based on a consensus about one or more measurement processes. This concept translates easily to MfI in the form of the ecosystem. Let us assume that for all $t \in T$ and $\lambda \in \Gamma_{cM}$, \exists an ecosystem chunk interpretation function c_E . The ecosystem chunk interpretation function determines the “true value” with respect to the ecosystem.

These ideas lead to definitions of accuracy. Suppose that c, c' are chunk interpretation functions, $t \in T$ and $\lambda \in \Gamma_{cM}$. Then we can say that:

- c is *accurate* (with respect to t and λ) if $c =_{\lambda t} c_E$;
- c is *inaccurate* (with respect to t and λ) if $c(\lambda, t) \cap c_E(\lambda, t) = \emptyset$;
- c is *more accurate than* c' (with respect to t and λ) if $c_E \leq_{\lambda t} c <_{\lambda t} c'$ or $c_E \geq_{\lambda t} c >_{\lambda t} c'$; (and similarly *less accurate*);
- c is *as accurate as* c' (with respect to t and λ) if $c_E \leq_{\lambda t} c \leq_{\lambda t} c'$ or $c_E \geq_{\lambda t} c \geq_{\lambda t} c'$;
- c is *partially accurate* if it is not accurate, inaccurate, or more or less accurate than c_E .

One interpretation has greater resolution than another if more chunks of content map to sets of slices. So, for any $t \in T, \Sigma' \in \Sigma$ define:

- $res(c, \Sigma', t) = |\{\lambda \in \Gamma_{cM}: \sigma \in c(\lambda, t), \text{ for some } \sigma \in \Sigma'\}|$;

- c has *greater resolution than* c' (with respect to t and Σ') if $\text{res}(c, \Sigma', t) > \text{res}(c', \Sigma', t)$;
- c has *lower resolution than* c' (with respect to t and Σ') if $\text{res}(c, \Sigma', t) < \text{res}(c', \Sigma', t)$;
- c has *resolution as good as* c' (with respect to t and Σ') if $\text{res}(c, \Sigma', t) \geq \text{res}(c', \Sigma', t)$;
- c has *the same resolution as* c' (with respect to t and Σ') if $\text{res}(c, \Sigma', t) = \text{res}(c', \Sigma', t)$.

Precision measures the variation in interpretation of a chunk over time. For any $\lambda \in \Gamma_{cM}$ define:

- the \cap -range of c with respect to λ , $\cap\text{-ran}(c, \lambda) = \{\sigma: \sigma \in c(\lambda, t) \forall t \in T\}$;
- the \cup -range of c with respect to λ , $\cup\text{-ran}(c, \lambda) = \{\sigma: \exists t \in T \text{ such that } \sigma \in c(\lambda, t)\}$;
- c is *precise* (with respect to λ) if $\cap\text{-ran}(c, \lambda) = \cup\text{-ran}(c, \lambda)$;
- c is *imprecise* (with respect to λ) if $\cap\text{-ran}(c, \lambda) = \emptyset$;
- c is *more precise than* c' (with respect to λ) if
 - $\cap\text{-ran}(c, \lambda) \supset \cap\text{-ran}(c', \lambda)$ and $\cup\text{-ran}(c, \lambda) \subseteq \cup\text{-ran}(c', \lambda)$, or
 - $\cap\text{-ran}(c, \lambda) \supseteq \cap\text{-ran}(c', \lambda)$ and $\cup\text{-ran}(c, \lambda) \subset \cup\text{-ran}(c', \lambda)$;
- c is *as precise as* c' (with respect to λ) if
 - $\cap\text{-ran}(c, \lambda) \supseteq \cap\text{-ran}(c', \lambda)$ and $\cup\text{-ran}(c, \lambda) \subseteq \cup\text{-ran}(c', \lambda)$.

An assertion consists of two chunks and a relationship. The definitions above show how accuracy, precision and resolution apply to chunks. What can be said of the relationship? Each member of Ξ is discrete and there is no implied partial order. So the only comparison that makes sense is equality.

Therefore, suppose that:

- a_1 is an assertion interpretation with induced chunk interpretations c_1 and c_1' and relationship r_1 ;
- a_2 is an assertion interpretation with induced chunk interpretations c_2 and c_2' and relationship r_2 ;
- $t \in T, \Sigma' \in \Sigma, \lambda \in \Gamma_{cM}$.

We can make the following definitions:

- a_1 is *accurate* (with respect to t and λ) if c_1 is accurate and c_1' is accurate and $r_1 = r_E$ (where r_E has the obvious definition);
- a_1 is *more accurate than* a_2 (with respect to t and λ) if
 - $r_1 = r_2$ and
 - c_1 is more accurate than c_2 and c_1' is as accurate as c_2' , or
 - c_1 is as accurate as c_2 and c_1' is more accurate than c_2' ;
- a_1 has *greater resolution than* a_2 (with respect to t and Σ')
 - $r_1 = r_2$ and
 - c_1 has greater resolution than c_2 and c_1' has resolution as good as c_2' , or
 - c_1 has resolution as good as c_2 and c_1' has greater resolution than c_2' ;
- a_1 is *more precise than* a_2 (with respect to λ) if
 - $r_1 = r_2$ and
 - c_1 is more precise than c_2 and c_1' is as precise as c_2' , or
 - c_1 is as precise as c_2 and c_1' is more precise than c_2' .

Precision applies to interpretations of the same content at different times. This may include multiple instances of the same piece of content in the same passage. In this case, as in others, the context may determine that the different interpretations should be different—that, according to the interpretation,

they are intended to refer to different sets of slices. In this case we need to narrow the measures to *groups* of content, that is content which, according to the rules of the MT are intended to refer to the same thing.

The largest unit to which IQ applies is the passage—a related sequence of assertions. Using the above definitions we can consider how the notion of quality applies to a passage and, in particular, to some adjectives that are sometimes applied. In [19] the authors say that accuracy, completeness, consistency and timeliness are “frequently mentioned dimensions” of IQ. Consider these in turn.

The accuracy of a passage depends on the accuracy of the individual assertions which comprise the passage.

Completeness is a more difficult topic. An IA can never be a complete description of a slice. This is the “blind men and the elephant” point discussed, for example, in [20] and [21]. So completeness does not fit well with intrinsic IQ. However, with action IQ the position is different. The question here is the following: is an IA sufficiently complete to enable the IE to choose reliably between different options for action? This question breaks down into two components:

- coverage: does the IA contain content which is derived from enough properties?
- resolution: does the interpretation provide sufficient resolution to discriminate between the options available?

Suppose that with respect to a particular MT, M:

- a chunk c is the subject of assertions a_1, \dots, a_n ;
- λ_n is the passage which contains a_1 to a_n in sequence.

Then we can say that λ_n has *coverage greater than or equal to* λ_{n-1} and *less than or equal to* λ_{n+1} . Again this is a partial order on passages induced by (reverse) set inclusion on the range of a passage interpretation.

Consistency addresses the question: is the interpretation of a given chunk (or atom) the same in different parts of the passage? Precision measures changes in interpretation so we can use precision to measure consistency.

Timeliness is a measure of a totally different type. It measures the gap between the time of measurement of properties and the time of interpretation.

Therefore, we can characterise the two types of IQ in the following way. Intrinsic IQ is a combination of accuracy, precision, resolution and timeliness. Action IQ adds in coverage as well.

4.4. Assessing IQ

The definitions above compare an IE’s interpretation with an ecosystem interpretation given a particular IA. However, if the process by which the IA was created can be examined then a further avenue is opened up. Any IA is created through a repeated process of sensing, interpretation and creation, potentially combining many different elements.

In theory, the measures described above can be applied to each stage of interpretation, but there is also a reverse process in which a set of connections inside an IE is converted into an IA. Again we can apply the measures above in exactly the same way as before to this position (only the time order has changed—we are still comparing an IE interpretation with an ecosystem interpretation).

So, theoretically, if there is sufficient evidence available, it is possible to examine the various processes at work to assess IQ. However, this can be a very resource intensive process to which different ecosystems have different approaches. The following Table 5 provides some examples.

Table 5. Assessing IQ—some examples.

Ecosystem	MT	Assessment of IQ	Characteristics
Science	Mathematics (often)	Careful design of experiments Peer review Transparency	High IQ Resource intensive
English Law	Legal English	Rigorous evidence gathering Trials and cross-examination	Relatively high IQ Resource intensive
Computer protocols	Tools like XML	Careful specification Careful implementation and testing Message validation	IQ linked to quality of definition Relatively resource intensive definition and implementation but efficient use (generally)
English speakers	Spoken English	Questions and answers	Variable IQ Relatively efficient

Because of the potential heavy resource overhead, it is natural that selection pressures have identified shortcuts that are good enough in many circumstances. However, nevertheless, since IQ is important, it is important that IEs can determine if the quality of an IA is good or not. Consider the following examples:

- “I saw it with my own eyes”: the implicit assumption here is that the senses are reliable (although there is much evidence that this is untrue—see [10] or [16] for examples);
- people trust brands for information just as they trust brands for other products (think of news organisations, for example);
- when users use a computer system they rely on the testing that has been carried out.

These examples imply that IQ not usually examined. Kahneman [10] shows this for humans with his description of System 1. In practice, it is too difficult for IEs to assess IQ so they often use proxies, including those in the list above. In [22] the authors express this idea elegantly (with respect to documents):

“For information has trouble, as we all do, testifying on its own behalf. Its only recourse in the face of doubt is to add more information. ... Piling up information from the same source doesn’t increase reliability. In general, people look beyond information to triangulate reliability.”

In these cases, the IE trusts that the IE providing the IA is reliable and that the reliability will transfer to the IA. This trust is not universal—it is an attribute of information ecosystems. The topic of trust in relation to IT is analysed in [23].

5. Information Friction (IF)

Selection pressures drive improvements in both IQ and efficiency. We can measure efficiency using the concept of information friction (IF). In [24], the authors were among many to talk of the frictionless economy, capitalism or information. Definitions vary and often include IQ as a component, but in this paper the term friction is used as a measure of the amount of resource used in measuring, transporting, converting and interpreting information. “Resource” is a general term which encompasses effort, food, fuel, equipment, expenses and so forth. To enable comparison we will assign a single value to resource usage. This is the approach taken in business in which all assets are assigned a (financial) value.

Suppose that:

- Φ is the set of all d-interaction function types;
- Ψ is the set of all space-time locations;
- N is the set of natural numbers;

Then we can define a friction function, $\epsilon: \Phi \times \Psi \rightarrow N$. The space-time location used corresponds to the relative space-time offset used in the properties of the d-interaction.

Friction applies along all parts of the derivation of an IA so, in turn, we will consider measurement friction, transmission friction, conversion friction and interpretation friction. In each case, friction is a primarily a pragmatic issue, not a theoretical one.

Measurement friction measures the cost of measuring the physical properties of slices. There is a large range of possibilities here ranging from the complicated and man-made (the Large Hadron Collider) with very high IF to the complicated and natural (human senses) with low IF.

Transmission friction has seen the largest impact of changes in technology.

Conversion changes the modelling tool used for an IA. Simple examples include:

- translation from one (human) language to another
- translation from one computer data structure to another (perhaps storing an incoming message in a database).

This requires resources like processing (e.g., brain or CPU) and storage (e.g., brain or disc). Conversion friction therefore measures the cost of these resources. Changes between ecosystems usually have high friction because the efficiencies which have evolved within ecosystems are bypassed.

Interpretation friction includes many of the examples which make working with information difficult. Examples include:

- interacting with a computer (in which both parties are trying to interpret the other!)
- deciphering an ancient document
- reading a news story.

So, for example, the degree of usability of computer systems is an input to the calculation of interpretation friction.

Improvements in technology have seen a large reduction in IF. Two centuries ago sending a message to the other side of the world required a ship. Now we can send an email. IQ has not

improved at the same rate. The discussion above clarifies this point. For the most part, technological changes have reduced transmission friction—they make it much easier to copy IAs from place to place. This does not, in itself, improve IQ except with respect to timeliness. This effect is exacerbated by Kahneman’s Systems 1 and 2. We are not instinctively aware of deficiencies in IQ so the selection pressures to improve it are smaller.

6. Floridi’s Questions

In [2], Floridi articulated a number of questions about information. This section examines the relationship with the following questions, which MfI tackles directly:

- (P1) “what is information?”;
- (P12) “the informational circle: how can information be assessed? If information cannot be transcended but can only be checked against further information—if it is information all the way up and all the way down—what does this tell us about our knowledge of the world?”.

6.1. What is Information?

Floridi poses the following question: “what is information?”. In [25], Dodig Crnkovic and Hofkirchner provide the following summary:

“Information can be understood as range of possibilities (the opposite of uncertainty); as correlation (and thus structure), and information can be viewed as code, as in DNA Furthermore, information can be seen as dynamic rather than static; it can be considered as something that is transmitted and received, it can be looked upon as something that is processed, or it can be conceived as something that is produced, created, constructed It can be seen as objective or as subjective. It can be seen as thing, as property or as relation. It can be seen from the perspective of formal theories or from the perspective of informal theories It can be seen as syntactic, as semantic or as pragmatic phenomenon, and it can be seen as manifesting itself throughout every realm of our natural and social world.”

MfI provides a response to this statement and shows how these different elements combine. Consider each element in turn.

“Range of possibilities (the opposite of uncertainty)”

IEs need to be able to recognise and discriminate states in the environment which can lead to more or less favourable outcomes. They need to be able to maximise the quality of the link between these states and possible actions and outcomes. Moreover, they need to convert the assessment into instructions which cause actions which increase the chance that the outcomes will be favourable.

Section 4 discusses these questions in the context of IQ. Each step requires an interpretation which links connections in working memory reliably to external slices and which enables accurate discrimination of states and outcomes. This analysis links MfI, for example, to the following definitions:

“information is the distinction that makes a difference” [26];

“information is the difference that makes a difference” [27].

“As correlation (and thus structure)”

Within MfI, linnets model structures and different information ecosystems use different modelling tools which embed some of these structures. The structures model many different types of connection and correlation.

“As code”

IAs have a structure (including “code” in the sense of the quote above) which forms part of one or more modelling tools. The structure and processing co-evolve in information ecosystems.

“Dynamic, rather than static”

All of MfI is based on the operation of physical processes so it is inherently dynamic. The Physical Connection Models and Extended Connection Models described in Section 3 provide the link to physical processes which underpins the other models.

“Transmitted and received”

MfI is based on d-interactions in which properties of an entity and its environment are updated. Under some circumstances (which is called transmission in Section 3) and under the right ecosystem conventions this includes transmission and reception in the normal sense.

“Processed, produced, created, constructed”

The generic IE model described throughout the paper shows that IAs are sensed, analysed, created and actuated by IEs.

“Objective or subjective”

In MfI interpretation plays a central role (see Section 4). By its nature, interpretation is particular to an IE and in this sense can be considered subjective (if the term can apply to inanimate objects). When an interpretation is common to an ecosystem or many ecosystems then it may be considered objective.

“Thing, as property or as relation”

Within MfI, information is about models of relations between properties of slices, and this is required for IEs to connect environment states to outcomes. So, information is inherently about things, properties and relations.

“Formal theories or informal theories”

MfI generalises the idea of theory to that of modelling tool. Section 4 describes modelling tools which can be formal or informal.

“Syntactic and semantic”

Modelling tools and the connections using them embody syntax. Other connections, especially the links to outcomes embody semantics.

We can also use MfI to explore some of the other facets of information. The literature (e.g., [2,11,28]) discusses the relationship between different types of information such as facts, data, knowledge and so forth. Within MfI, these are all related to combinations of the following:

- the information ecosystem(s) under consideration;
- information quality;
- the degree of connection implied in an IA or created in an IE.

For example, we can consider facts to be information for which IQ is high for a large number of ecosystems. Knowledge resides in the connections created in the working memory of IEs between

environment states and outcomes. This is consistent with the position of Lenski in [28] in which he says: “knowledge is a linked structure of concepts.”

MfI also reinforces the view that information is relative. In [11], Burgin says that: “the information in a message is a function of the recipient as well as the message.” Logan reviews the relativity of information in [9] and concludes that: “information is not an absolute but depends on the context in which it is being used.” This idea is embedded in MfI. The generic IE model and the discussion about interpretation and connections in working memory make this point clear.

We can also apply MfI to the word “information” itself. Just like any other word, the interpretation of the word “information” is ecosystem- and IE-dependent (which is, in itself, an explanation for the numerous definitions of information [2,11]). It is clear from the IQ discussion that modelling tools are more or less appropriate for different purposes; language is not appropriate if high levels of accuracy, precision and resolution are required (this is why science uses mathematics). This implies that a language-based definition of information may be more open to different interpretations than definitions using more rigorous modelling tools. One of the lessons of MfI is that the right set of modelling tools is required for any purpose.

Information is relative in a different sense—it depends on what is measured, how it is measured and the processes which deliver the information to an IE. It is impossible to capture the entire state of a slice (see [20,21]). However, once the information is measured it may be sensed, interpreted, combined and transmitted many times. The IQ measures discussed in Section 4 provide a mechanism for analysing this further. One conclusion is that two identical IAs may have different IQ when both are assessed (for example, the same words from an expert or a neophyte).

6.2. How can Information be Assessed?

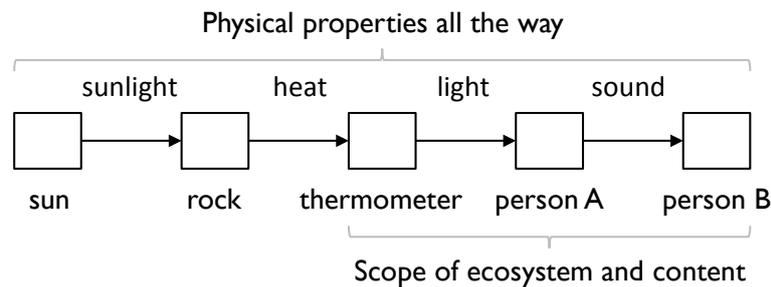
In [2], Floridi poses the following question:

“The informational circle: how can information be assessed? If information cannot be transcended but can only be checked against further information—if it is information all the way up and all the way down—what does this tell us about our knowledge of the world?”

The different MfI models show how information is linked directly to the properties of physical entities. Consider, in turn, the different elements of this question.

“If information cannot be transcended but can only be checked against further information— if it is information all the way up and all the way down...”

In MfI, the relationship between physical properties and content is the distinction between PCMs and IAMs (discussed in Section 3), and the development of these elements of MfI shows the relationship. As an example, consider sunlight shining on a rock to which a thermometer is attached, which person A reads and tells person B, as in the following diagram (Figure 5).

Figure 5. Physical properties and content.

In this example, an ecosystem convention defines the point at which properties become content.

Content is connected to the properties of physical content in two ways. As defined and analysed in Section 3, content is a set of physical properties which (according to ecosystem conventions and processes) is processed in a particular way. Content is also derived from the properties of other physical entities. This can happen in the following ways:

- through interaction (as in the example above);
- through processing existing content within an IE, in which case the new content is derived from the properties of the physical entities which represent the existing content within the IE by some physical processes;
- through the creation of the IE, in which case it is based on a set of physical processes acting on a pattern (which again comprises some properties of physical entities—see Section 2).

Some of the examples created through processing form an interesting category. They include falsehoods, syntactically correct nonsense or imagined ideas—in each case not directly linked to external entities. They are still created in the working memory of an IE by linking content and are properties of physical entities but the difference is that they have very low (or undefined) IQ.

So, according to MfI, instead of the statement above it is more correct to say:

“It is properties of physical entities all the way up and all the way down and, according to ecosystem conventions, some are treated as content.”

Now consider the following question:

“What does this tell us about our knowledge of the world?”

According to MfI the premise for this is incorrect, so the question as posed becomes immaterial. However, MfI shines some light on this question when separated from the premise. As described in Section 4, information purports to describe set theoretic relationships between slices characterised using different combinations of properties. This purpose is subject to the vagaries of IQ and, as a result, is delivered with varying degrees of success. This purpose enables IEs to link environment states with potential outcomes in order to help the IE navigate towards a relatively favourable outcome.

7. Conclusions

This paper shows that a model for information which integrates evolutionary, static and dynamic viewpoints provides a richer understanding of the nature of information than a focus on definitions alone and, as a result, provides a sound basis for examining some important questions about information.

The model discusses modelling tools in relation to information. Modelling tools such as language and Mathematics have their advantages and disadvantages. High IQ is difficult with language. Mathematics provides high IQ but at the cost of high IF. This prompts some questions: what other forms of modelling tool strike a different balance? Can we use them to understand information itself better? The application of the tools of MfI to these questions in addition to some of Floridi's other questions points to directions for further research.

Acknowledgments

Many thanks are due to the referees for their constructive comments.

Conflicts of Interest

The author declares no conflict of interest.

References

1. Silver, G.; Silver, M. *Systems Analysis and Design*; Addison Wesley: Reading, MA, USA, 1989.
2. Floridi, L. *The Philosophy of Information*; Oxford University Press: Oxford, UK, 2011.
3. Del Moral, R.; Navarro, J.; Marijuán, P. New Times and New Challenges for Information Science: From Cellular Systems to Human Societies. *Information* **2014**, *5*, 101–119.
4. Hofkirchner, W. *Emergent Information: A Unified Theory of Information Framework*; World Scientific Publishing: Singapore, Singapore, 2013.
5. Burgin, M. Evolutionary Information Theory. *Information* **2013**, *4*, 124–168.
6. Darwin, C. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*; John Murray: London, UK, 1859.
7. Smith, A. *An Inquiry into the Nature and Causes of the Wealth of Nations*; Strahan and Cadell: London, UK, 1776.
8. Avgeriou, P.; Uwe, Z. Architectural patterns revisited: A pattern language. In Proceedings of 10th European Conference on Pattern Languages of Programs (EuroPlop 2005), Bavaria, Germany, July 2005.
9. Wolff, J. The SP Theory of Intelligence: Benefits and Applications. *Information* **2014**, *5*, 1–27.
10. Kahneman, D. *Thinking, Fast and Slow*; Macmillan: London, UK, 2011.
11. Burgin, M. *Theory of Information: Fundamentality, Diversity and Unification*; World Scientific Publishing: Singapore, Singapore, 2010.
12. Harary, F. *Graph Theory*; Addison-Wesley: Reading, MA, USA, 1969.
13. Harary, F.; Gupta, G. Dynamic Graph Models. *Math. Comput. Model.* **1997**, *25*, 79–87.

14. ISO/IEC 9075-11:2008: Information technology—Database languages—SQL—Part 11: Information and Definition Schemas (SQL/Schemata). Available online: http://www.iso.org/iso/catalogue_detail.htm?csnumber=38645 (accessed on 15 September 2014).
15. Tulving, E. Episodic and semantic memory. In *Organization of Memory*; Tulving, E., Donaldson, W., Eds.; Academic Press: New York, NY, USA, 1972; pp. 381–403.
16. Macknik, S.L.; Martinez-Conde, S. *Sleights of Mind*; Picador: Surrey, UK, 2011.
17. Kohlas, J.; Schmid, J. An Algebraic Theory of Information: An Introduction and Survey. *Information* **2014**, *5*, 219–254.
18. Popper, K.R.; *The Logic of Scientific Discovery*; Hutchinson: London, UK, 1959.
19. Huang, K.-T.; Lee, Y.W.; Wang, R.Y. *Quality Information and Knowledge*; Prentice Hall: Upper Saddle River, NJ, USA, 1999.
20. Fields, C. Equivalence of the Symbol Grounding and Quantum System Identification Problems. *Information* **2014**, *5*, 172–189.
21. Brenner, J. Information: A Personal Synthesis. *Information* **2014**, *5*, 134–170.
22. Brown, J.S.; Duguid, P. *The Social Life of Information*; Harvard Business Press: Boston, MA, USA, 2000.
23. DeVries, W. Some Forms of Trust. *Information* **2011**, *2*, 1–16.
24. Gates, B.; Myhrvold, N.; Rinearson, P. *The Road Ahead*; Viking Penguin: New York, NY, USA, 1995.
25. Dodig Crnkovic, G.; Hofkirchner, W. Floridi's "Open Problems in Philosophy of Information", Ten Years Later. *Information* **2011**, *2*, 327–359.
26. Mackay, D.M. *Information, Mechanism and Meaning*; MIT Press: Cambridge, MA, USA, 1969.
27. Bateson, G. *Steps to an Ecology of Mind*; Ballantine Books: New York, NY, USA, 1972.
28. Lenski, W. Information: A Conceptual Investigation. *Information* **2010**, *1*, 74–118.

© 2014 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).