SEENG 609

Theoretical Foundations of Software Engineering

Lecture 6

Cognitive Informatics Foundations of Software Engineering

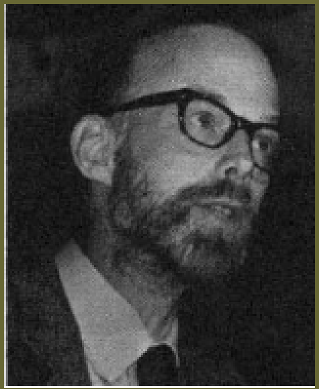
Yingxu Wang, *Prof., PhD, PEng, FWIF, SMIEEE, SMACM*

Visiting Professor: Stanford Univ. (2008), UC Berkeley (2008), Oxford Univ. (1995)
Director, Theoretical & Empirical Software Eng. Research Centre (TESERC)
Dept. of Electrical and Computer Engineering
Schulich School of Engineering, University of Calgary
Office: ICT524, Tel: 220 6141
Email: yingxu@ucalgary.ca
http://www.enel.ucalgary.ca/People/wangyx/

1. Introduction

Discussions on C.A.R. Hoare

Read the classic paper written by C.A.R. Hoare:



C.A.R. Hoare (1980), *The Emperor's Old Clothes*, The 1980 Turing Award Lecture, Communications of the ACM, Vol.24, No.2, pp.75-83.

Prepare a 6-minute group talk in next lecture on the following topics:

- About the author
- What are the 'new clothes' of the emperor in SE? Provide an example of promising technical hoaxes that was popular but disappeared shortly in SE.
- What conclusions of the article interested you? Why?
- Your argument(s) or count-points on any of the conclusions derived in this article.

Review of Lecture 5

Lecture 5. Systems Science Foundations of SE

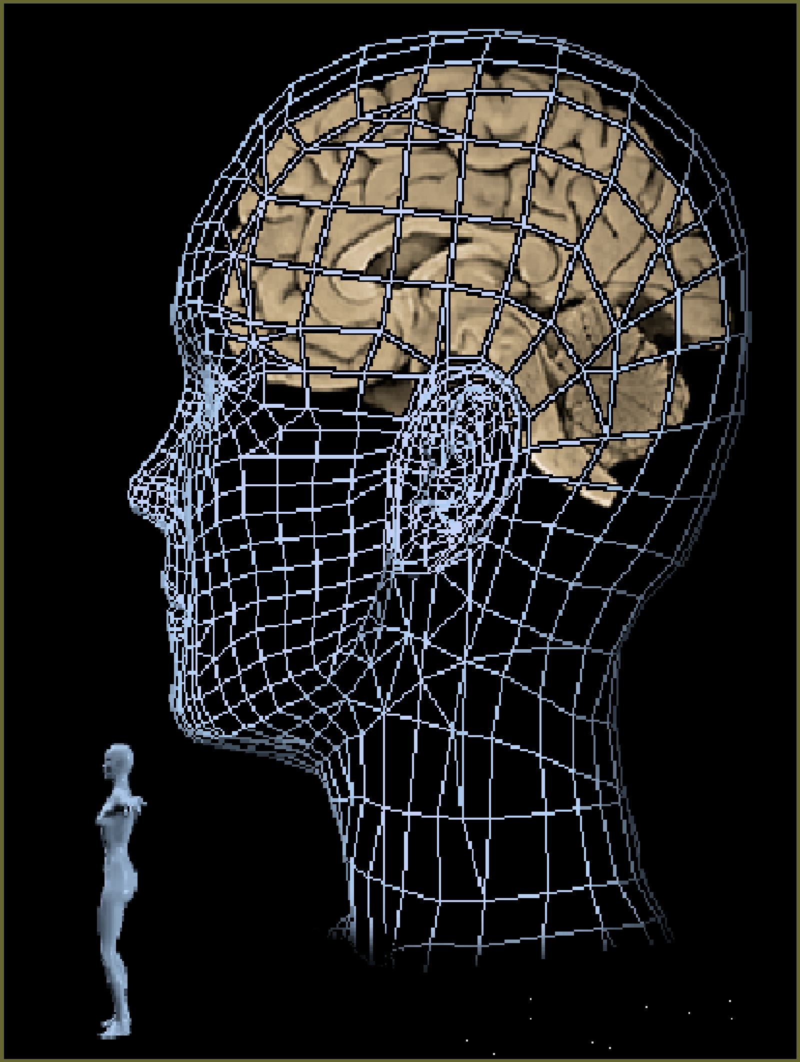
- **Systems**
  - Classic systems theories
  - Contemporary systems theories
  - Systematic thinking
  - System philosophies
- **System topology**
  - Taxonomy of systems
  - Sizes and magnitudes of systems
  - Hierarchical architectures of systems
- **System algebra**
  - Mathematical models of *abstract systems*: closed/open systems
  - *System relations:* equivalence / independence / subsystem / super system
  - *System operations:* reproductive (4); compositional (4)
- **Principles of system science**
  - System fusion
  - System functions / behaviors
  - The law of maximum system gains
  - System equilibrium
  - System dissimilation
  - System mutation
  - Abstract work done by systems
  - Systems synchronization
  - System self-organization
- **Software system engineering**
  - The abstract model of software systems
  - Hierarchical structures of SE work products
  - System engineering perspectives on SE

Brain and NI – The Last World Yet to be Explored

- **Natural Intelligence**

  - In the *narrow sense*, is a human or a system ability that transforms information into behaviors;

  - In the *broad sense*, is any human or system ability that autonomously transfers the forms of abstract information between *data, information, knowledge,* and *behaviors* in the brain.



A Profound Problem in CS and SE

- A fundamental question in software science:

  - Software is not constrained by any laws and principles discovered in the physical world!

  - What are the constraints that software obeys?

2. From Classic and Modern Information Theories to Cognitive Informatics

1. Introduction

➔ **2. From classic and modern information theories to cognitive informatics**

3. Cognitive informatics and the brain

4. Cognitive informatics for SE

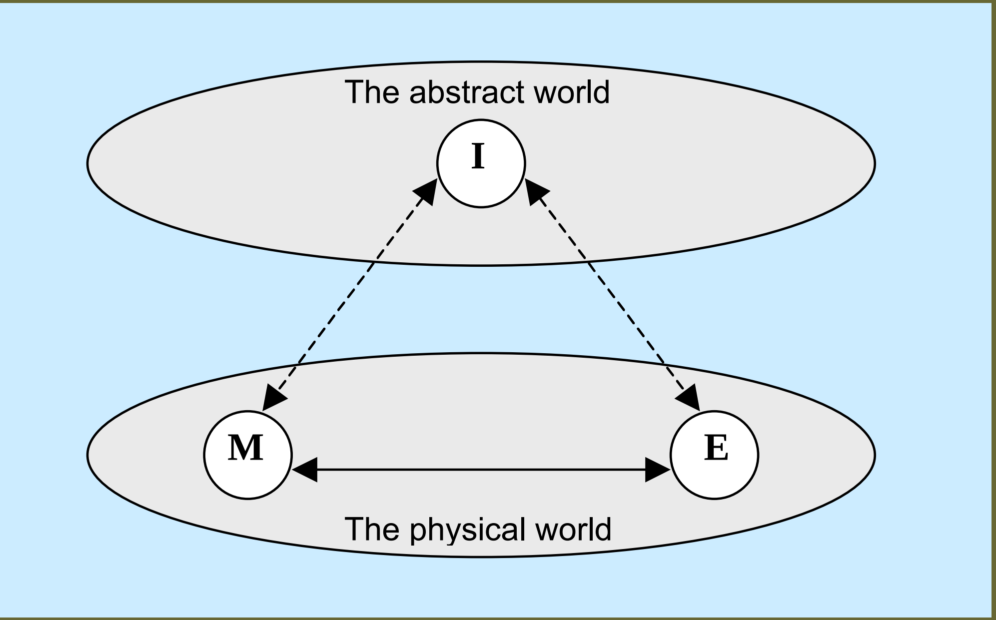
5. Cognitive complexity of software

6. Summary

7. Assignment

Information – The Third Essence of the Natural World

- Information is the 3rd essence in modeling the world.
- Any product and/or process of human mental activities results in information.
- Information, matter, and energy may be transferred between each other.
- Software obeys the laws of modern informatics and cognitive informatics.



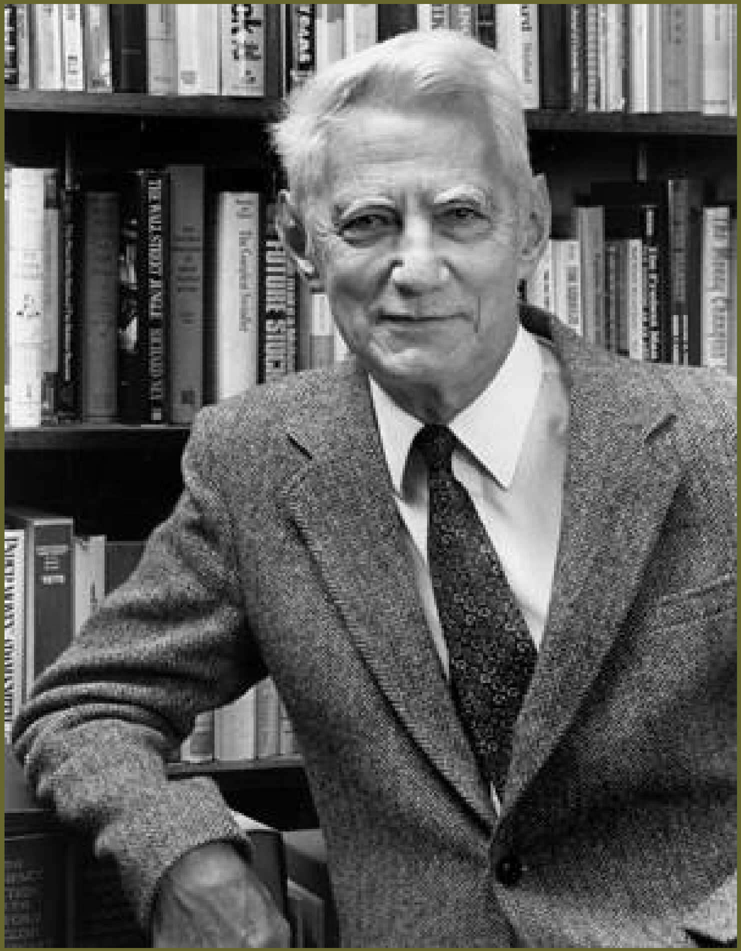
The Primitive Form of Information & Computing

- **Theorem 6.1**

The *primitive form of information* states that the most fundamental form of information that can be represented and processed is binary digit where *k* = *b* = *2*, i.e.:

$$
\begin{aligned}
I_b &= f : M \to S_b \\
&= \lceil \log_b M \rceil \\
&= \lceil \log_2 M \rceil \quad [\text{bit}]
\end{aligned}
$$

2.1 Classical Informatics



(1916 - 2001)

- Shannon's work founded the subject of Information Theory.

Refer to:

[1] Shannon, C.E. [1948], A Mathematical Theory of Communication, *Bell System Technical Journal*, Vol.27, pp.379-423 and 623-656.

[2] Shannon, C.E. and Weaver, W. [1949], The Mathematical Theory of Communication, *Illinois University Press,* Urbana, IL, USA.

Shannon's Definition of Information

- **Conventional Information Theory (Shannon, 1948)**

  - Definition 1: *Information* is defined as a probabilistic measure of the quantity of message which can be obtained from a message source.

  - Definition 2: *Information* is any kind of signals that can be sent via transmission channels.

    - Based on the probability theory
    - Focused on information transmission rather than information itself
    - The measure of the quantity of information is depend on the receiver's subjective judgment.

  - Shannon surprised the *communication theory* community by proving that this was not true as long as the communication rate was below channel capacity. The capacity can be simply computed from the noise characteristics of the channel.

Measurement of Information

- The ***average information content***, I_i, of the *i*th sign in a message is determined by its unexpectedness, i.e.:

$$I_i = \sum_{i=1}^{n} \log_2 (1/p_i) \quad \text{[bit]}$$

where p_i is the probability that the *i*th sign is transmitted.

- The unit of information is ***bit***, shortened from 'binary digit.'

Measurement of Information (Cont'd)

- The *total average information* transmitted by a source or sender, *I,* is the weighted sum of its *n* possible signs in the message, i.e.:

$$I = \sum_{i=1}^{n} p_i \bullet I_i$$

$$= \sum_{i=1}^{n} p_i \bullet \log_2 (1/p_i)$$

$$= \sum_{i=1}^{n} - p_i \bullet \log_2 p_i \qquad \text{[bit]}$$

- *Entropy* E is the extent of the trend of a system towards complete disorder or randomization. (*E = I*)

Measurement of Information (Cont'd)

- **E.g.: For a binary source that has an alphabet of two equally likely signs, or $p_i = 0.5$, its total average information, *I*, is:**

$$I = \sum_{i=1}^{n} p_i \bullet \log_2 (1/p_i)$$

$$= \sum_{i=1}^{2} 0.5 \bullet \log_2 (1/0.5)$$

$$= 1 \text{ [bit]}$$

- **It is noteworthy that for the above binary system, the average content of information is always 1 [bit], or generally *I* is not proportional to the sizes of messages.**

That is, on the basis of the classical information measurement, no matter how many bits message have been transmitted, the value of average information will not change for a given transmission system.

2.2 Contemporary Informatics

- Conventional informatics treats ***information*** **as a** ***probabilistic measure*** **of the quantity of message which can be received from a message source.**

  It was focused on information transmission rather than information itself.

- The modern informatics tends to regard information as ***entities of messages*****, rather than a measurement of messages in the classical information theory.**

  The new perception is found better to explain the theories and practices in the IT and IS industry.

Information: A New Perspective

- **Definition:** ***Information*** **in CI is defined as the generic abstract artefacts that can be modeled, processed, and stored by human brains.**

  - with this orientation, information is regarded as the entity of messages, rather than a measurement of messages in the classical information theory.

  - The nature of information in human brain

  - Internal information processing
  acquisition, memory, categorization, retrieve, and generation

Measurement of Information

- The *content of information* in modern informatics is measured by the cost of code to abstractly represent a given size of message *M* in a digital system *k*, i.e.:

$$I_k = f: M \to S_k$$
$$= \log_k M$$

where

- I_k, the content of information in digital system *k*
- S_k, the measurement scale based on *k*.

Measurement of Information (Cont'd)

- When a binary digital representation system is adopted, i.e. $k = b = 2$, the content of information becomes the most useful one:

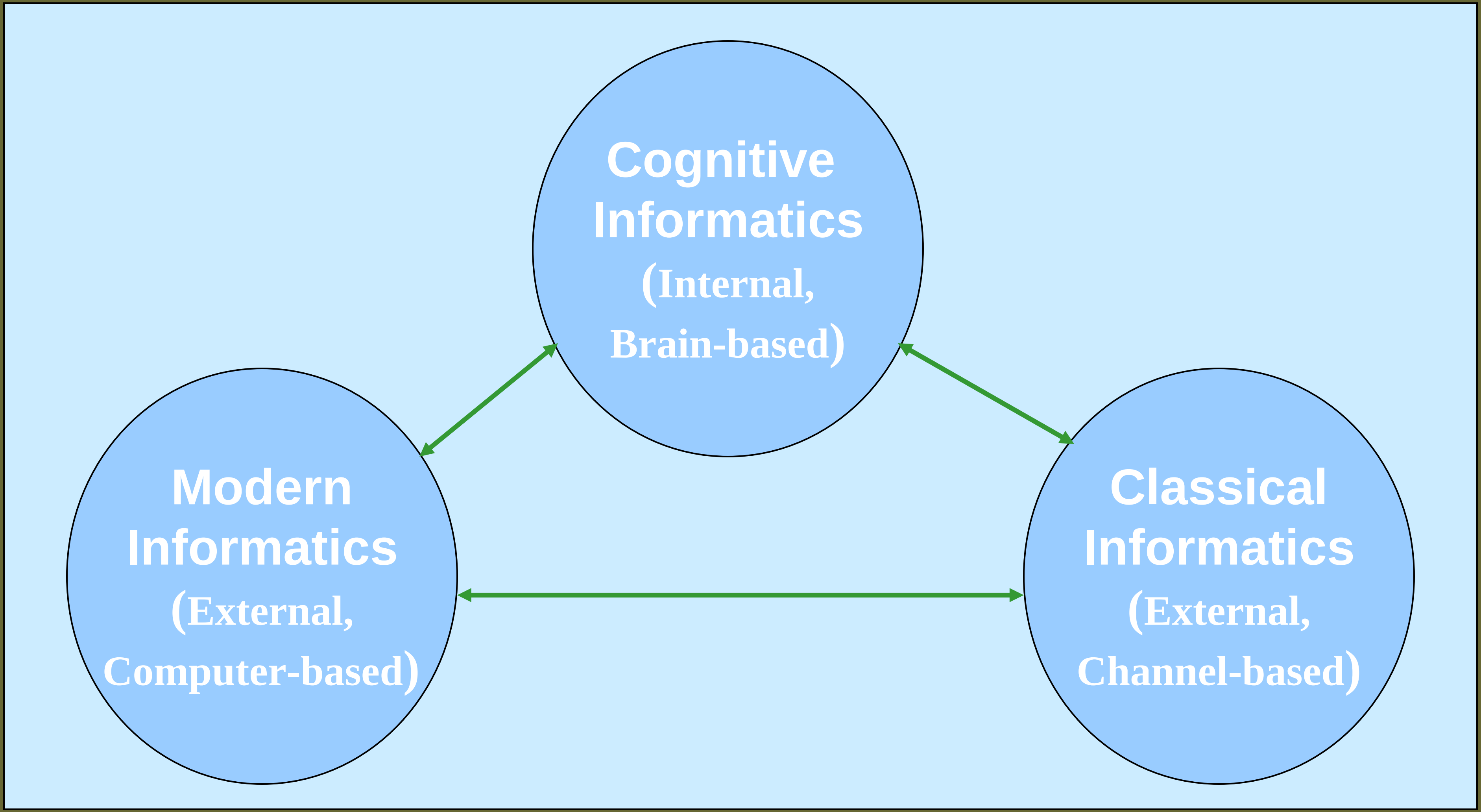
$$I_b = f: M \rightarrow S_b$$
$$= \log_2 M \qquad \text{[bit]}$$

where the unit of information, I_b, is a *bit*.

- Note the bit here is concrete and deterministic, and no longer possesses a property as a value of weighted probability as that in the classical informatics.
- E.g.: For a given message M = 2^{300} bits, its information contents can be determined as follows:

$$I_b = \log_2 M$$
$$= \log_2 2^{300}$$
$$= 300 \text{ [bit]}$$

The Relationship between External & Internal Informatics

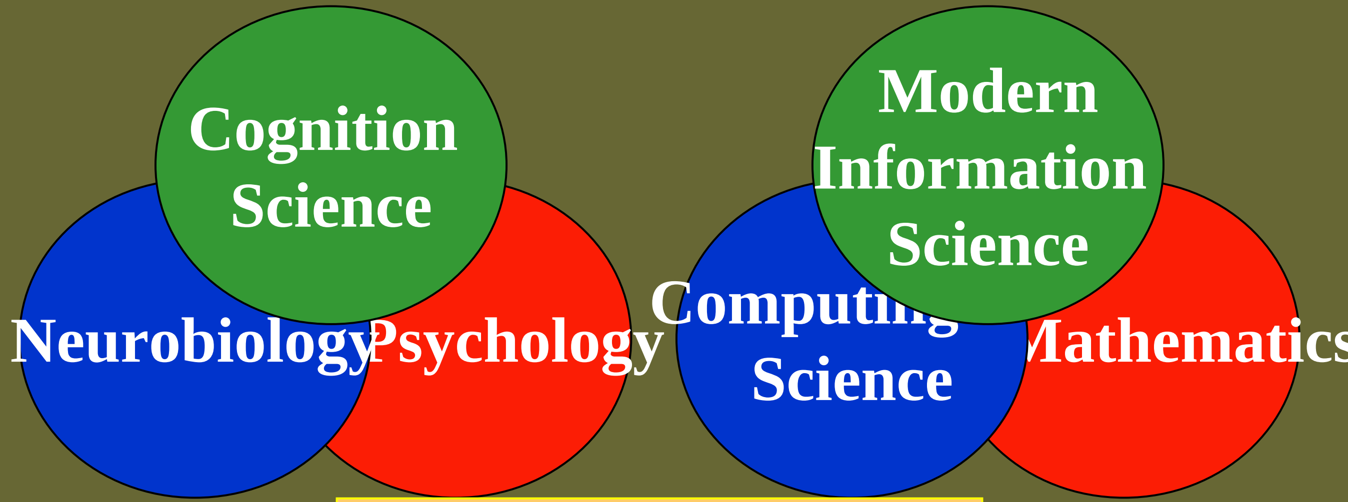


3. Cognitive Informatics and the Brian

1. Introduction
2. From classic and modern information theories to cognitive informatics
➔ **3. Cognitive informatics and the brain**
4. Cognitive informatics for SE
5. Cognitive complexity of software
5. Summary
7. Assignment

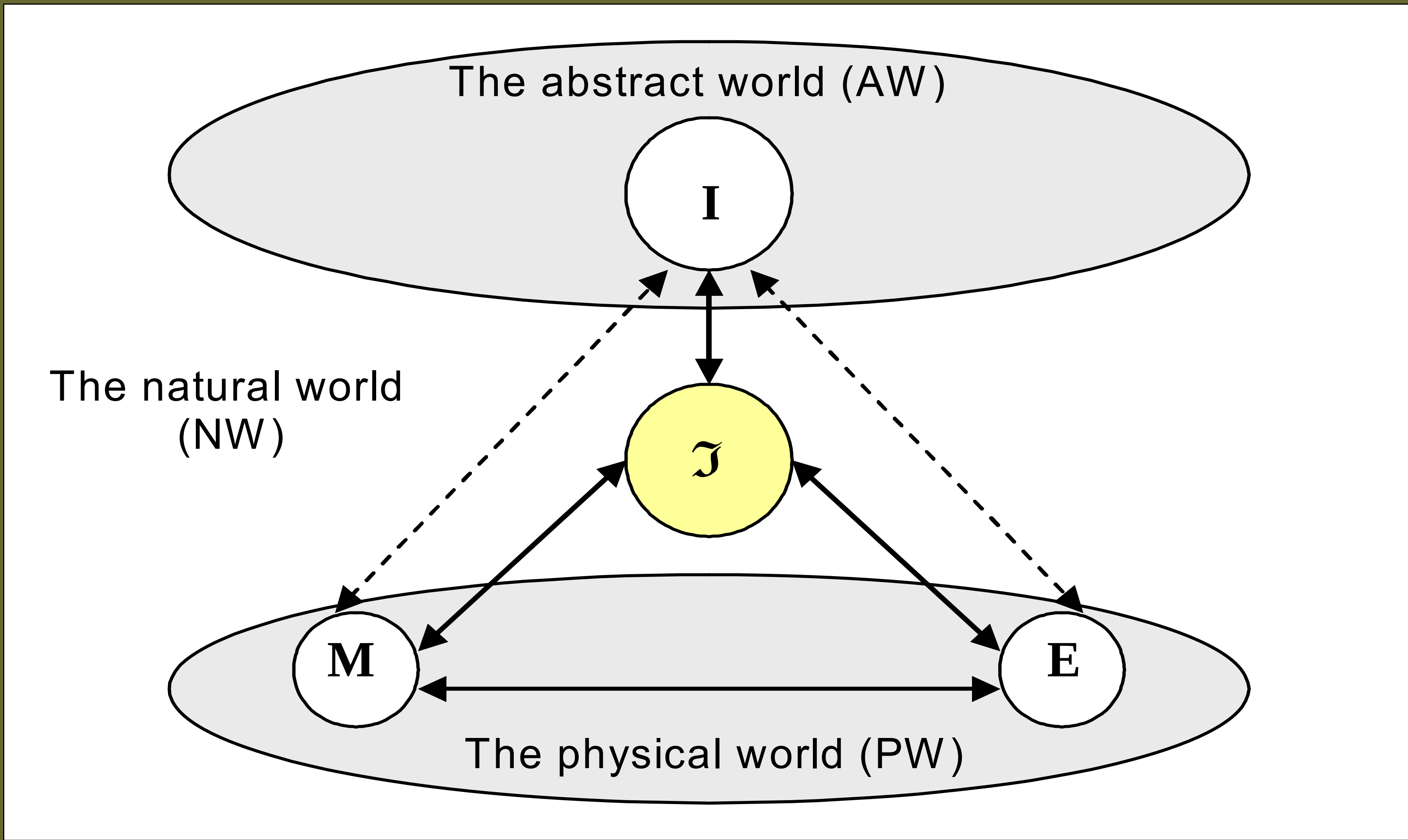
A New Frontier: Cognitive Informatics (CI)

- ***Cognitive informatics* (CI)** **is a transdisciplinary enquiry of cognitive, computing, and information sciences, which studies the internal information processing mechanisms and processes of *natural intelligence* (the brain), the theoretical framework and denotational mathematics of *abstract intelligence*, and their engineering applications by *cognitive computing*.**



Cognitive Informatics (CI)

3.1 Abstract Intelligence (αI)

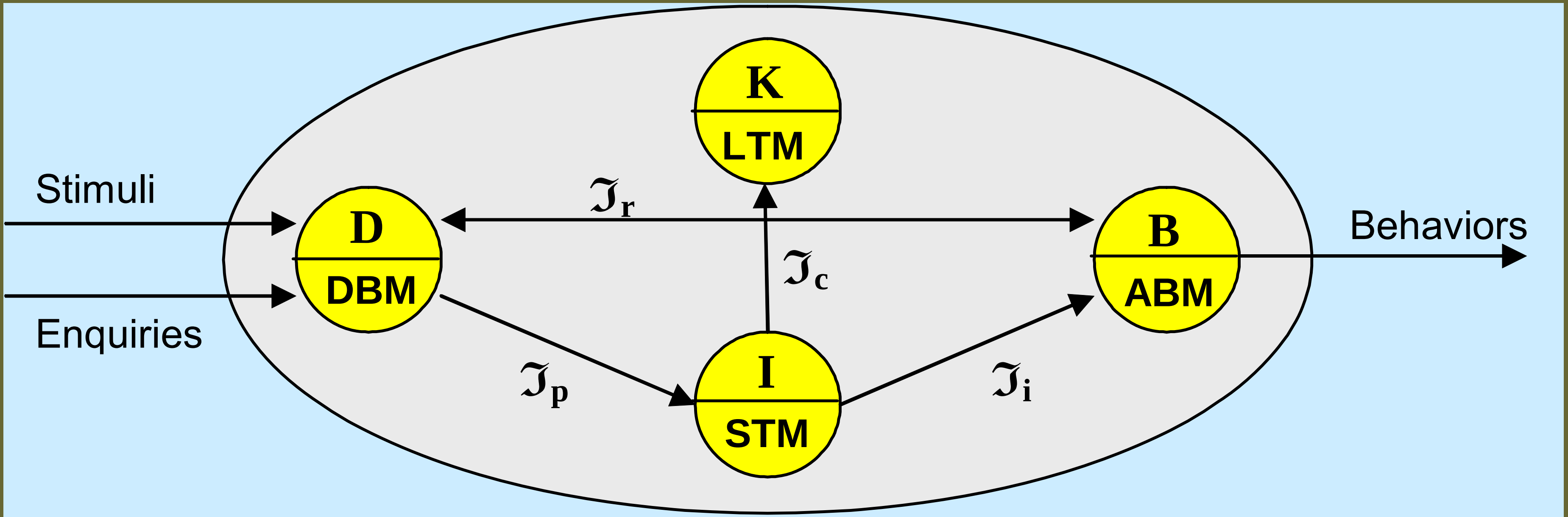


αI and Paradigms

- ***Abstract intelligence*, αI,** **is a human enquiry of both natural and artificial intelligence at the embody levels of neural, cognitive, functional, and logical from the bottom up.**

No.	Form of intelligence	Embodying Means	Paradigms
1	Natural intelligence (NI)	Naturally grown biological and physiological organisms	Human brains and brains of other well developed species
2	Artificial intelligence (AI)	Cognitively-inspired artificial models and man-made systems	Intelligent systems, knowledge systems, decision-making systems, and distributed agent systems
3	Machinable intelligence (MI)	Complex machine and wired systems	Computers, robots, autonomic circuits, neural networks, and autonomic mechanical machines
4	Computational intelligence (CoI)	Computational methodologies and software systems	Expert systems, fuzzy systems, autonomous computing, intelligent agent systems, genetic/evolutionary systems, and autonomous learning systems

The Generic Abstract Intelligence Model (GAIM)



The Natural Intelligence Model

$\mathfrak{I}_p$ – ***Perceptive*** **intelligence**

$\mathfrak{I}_c$ – ***Cognitive*** **intelligence**

$\mathfrak{I}_i$ – ***Instructive*** **intelligence**

$\mathfrak{I}_i$ – ***Reflective*** **intelligence**

Advances of the Human Brains

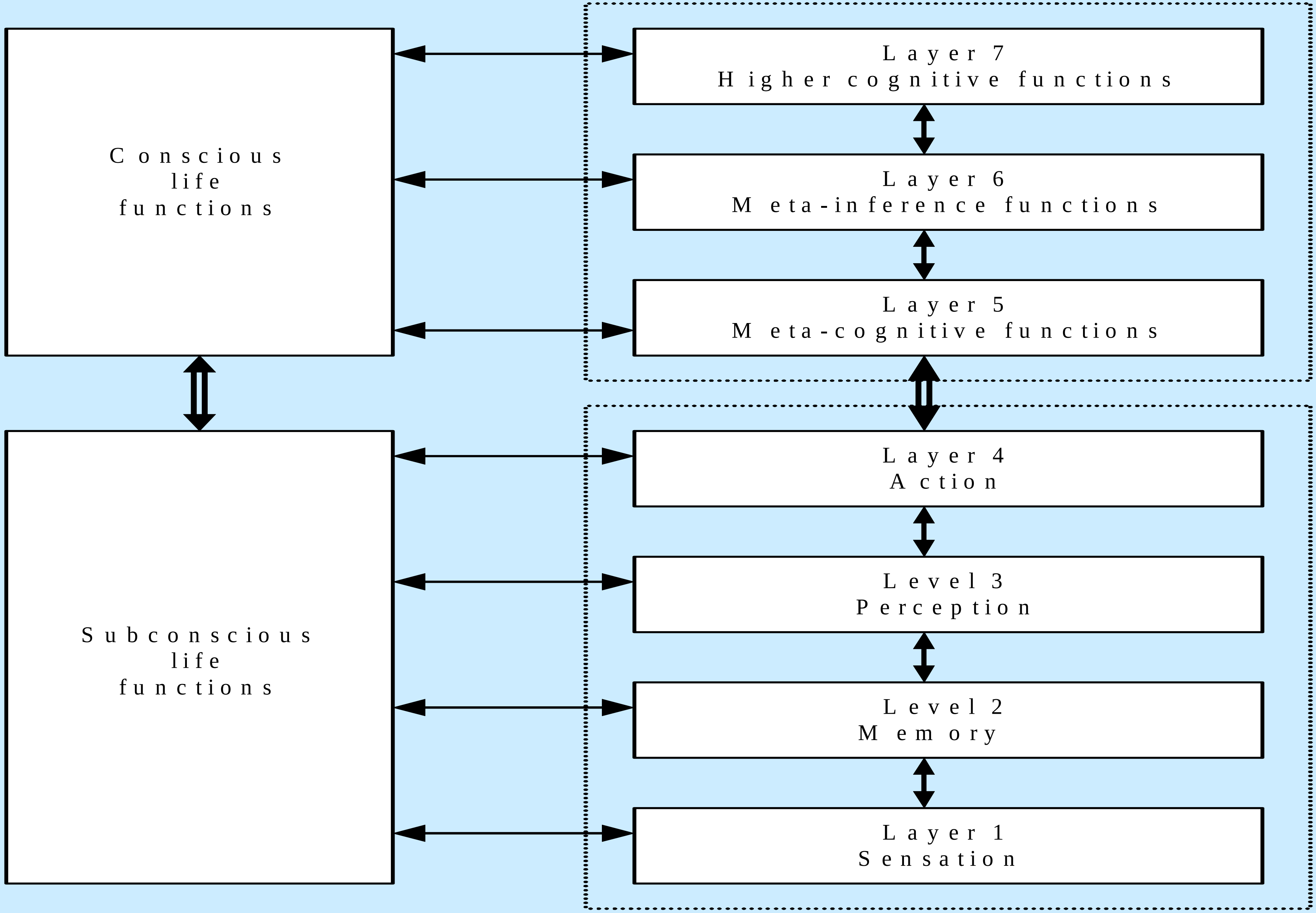
- **Theorem 6.2**

  The *quantitative advantage of human brain* states that the magnitude of the memory capacity of the brain is tremendously larger than that of the closest species.

- **Theorem 6.3**

  The *qualitative advantage of human brain* states that the possession of the abstract layer of memory and the abstract reasoning capacity makes human brain profoundly powerful on the basis of the quantitative advantage.

3.2 The Layered Reference Model of the Brain (LRMB)



The Cognitive Processes of the Brain

Life behaviors and complex actions

Layer 7: The higher cognitive processes

Comprehension	Learning	Problem solving	Decision making	Creation	Planning	Pattern recognition

Layer 6: Meta inference processes

Deduction	Induction	Abduction	Analogy	Analysis	Synthesis

Layer 5: Meta cognitive processes

Object Identify	Abstra-ction	Concept establish.	Categori-zation	Compa-rison	Memori-zation	Qualifi-cation	Quantifi-cation	Selection	Search	Model establish.	Imagery

Layer 4: Action processes

Wired actions (Skills)	Contingent actions (Temporary behaviors)

Layer 3: Perception processes

Self-Consciousness	Attention	Motivation and goal-setting	Emotions	Attitudes	Sense of spatiality	Sense of motion

Layer 2: Memory processes

Sensory buffer Memory	Short-term Memory	Long-term Memory	Action buffer Memory

Layer 1: Sensational processes

Vision	Audition	Smell	Tactility	Taste

The physiological/neurological Brain

Classification of Cognitive Processes in LRMB

- The *subconscious* layers of the brain
  - *Inherited, fixed*, and relatively mature when a person was born.
  - The subconscious function layers are not directly controlled and accessed by the conscious life function layers.
  - Known as *non*conscious functions in some literature.
- The *conscious* layers of the brain
  - *Acquired* and *highly plastic*
  - Can be controlled intentionally based on willingness, goals, and motivations.

Usage of LRMB

- LRMB may be applied:
  - To explain a wide range of physiological, psychological, and *cognitive phenomena* in cognitive informatics
  - To explore the relationships between the *inherited* and the *acquired* life functions, as well as the *subconscious* and *conscious* cognitive processes.
  - To gain insight in designing and implementing ATS.

4. Cognitive Informatics for SE

1. Introduction
2. From classic and modern information theories to cognitive informatics
3. Cognitive informatics and the brain
4. → **4. Cognitive informatics for SE**
5. Cognitive complexity of software
6. Summary
7. Assignment

Laws and Properties of Information

(1) Abstraction
(2) Generality
(3) Cumulativeness
(4) Dependency on cognition
(5) Three-dimensional behavioral space
(6) Sharability
(7) Dimensionless
(8) Weightless
(9) Transformability between I-M-E
(10) Multiple representation forms
(11) Multiple carrying media
(12) Multiple transmission forms
(13) Dependability on media
(14) Dependability on energy
(15) Wearless and time dependency
(16) Conservation of information and thermal entropy
(17) Special quality attributes
(18) Susceptible to distortion
(19) Scarcity

(Wang, 2001/2005)

Cognitive Computing

- **Cognitive Computing**
  - *Cognitive computing* is an emerging paradigm of intelligent computing methodologies and systems that implements computational intelligence by autonomous inferences and perceptions mimicking the mechanisms of the brain.

- **Cognitive Computers**
  - A *cognitive computer* is an intelligent knowledge processor with the capabilities of autonomic inference and perception that mimics the mechanisms of the brain.

An Abstract Model of Cognitive Computers

The Computational Intelligent Model of CC (1/2)

§CC**ST** $\triangleq$ CCOS**ST** // CC operating system
|| CCIB**ST** // CC intelligent behaviors
= { // CCOS**ST**

<SE**ST**: $\underset{ptrP=0}{\overset{n-1}{R}}$ SENSORS[ptr**P**]**ST**> // Layer 1: Sensation engine

|| <ME**ST**: $\underset{addrP=0}{\overset{5}{R}}$ MEM[addr**P**]**ST**> // Layer 2: Memory engine
= <SBM**ST** || STM**ST** || CSM**ST** || LTM**ST** || ABM**ST**>

|| <PE**ST**: $\underset{iP=0}{\overset{7}{R}}$ PROC[i**N**]**ST**> // Layer 3: Perception engine
= <Attention**ST** || Motivation**ST** || Emotion**ST** || Attitude**ST**
|| SensOfSpatiality**ST** || SensOfTime**ST** || SensOfMotion**ST**>

|| <AE**ST**: $\underset{ptrP=0}{\overset{n_{SERVOH}-1}{R}}$ SERVOS[ptr**P**]**ST**> // Layer 4: Action engine

|| <CE**ST**: $\underset{iP=0}{\overset{10}{R}}$ PROC[i**N**]**ST**> // Layer 5: Meta-cognition engine
= <ObjectIdentification**ST** || Abstraction**ST** || ConceptEstablishment**ST**
|| Search**ST** || Categorization**ST** || Comparison**ST** || Memorization**ST**
|| Qualification**ST** || Quantification**ST** || Selection**ST**>

|| <IE**ST**: $\underset{iP=0}{\overset{6}{R}}$ PROC[i**N**]**ST**> // Layer 6: Meta-inference engine
= <Deduction**ST** || Induction**ST** || Abduction**ST** || Analogy**ST**
|| Analysis**ST** || Synthesis**ST**>

|| <HCE**ST**: $\underset{iP=0}{\overset{7}{R}}$ PROC[i**N**]**ST**> // Layer 7: Higher cognition engine
= <Comprehension**ST** || Learning**ST** || Planning**ST** || ProblemSolving**ST**
|| DecisionMaking**ST** || Creation**ST** || PattenRecognition**ST**>
|| <§t**TM**> // Relative clock
}

An Abstract Model of Cognitive Computers

The Computational Intelligent Model of CC (2/2)

|| { // CCIB**ST**

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_e\mathbf{N}-1} @ e_k\mathbf{S} \hookrightarrow P_k\mathbf{ST}>$$ // *Event*-driven behaviors (B_e)

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_t\mathbf{N}-1} @ t_k\mathbf{TM} \hookrightarrow P_k\mathbf{ST}>$$ // *Time*-driven behaviors (B_t)

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_{int}\mathbf{N}-1} @ int_k\odot \hookrightarrow P_k\mathbf{ST}>$$ // *Interrupt*-driven behaviors (B_{int})

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_t\mathbf{N}-1} @ g_k\mathbf{ST} \hookrightarrow P_k\mathbf{ST}>$$ // *Goal*-driven behaviors (B_g)

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_t\mathbf{N}-1} @ d_k\mathbf{ST} \hookrightarrow P_k\mathbf{ST}>$$ // *Decision*-driven behaviors (B_d)

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_t\mathbf{N}-1} @ p_k\mathbf{ST} \hookrightarrow P_k\mathbf{ST}>$$ // *Perception*-driven behaviors (B_p)

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_{int}\mathbf{N}-1} @ inf_k\mathbf{ST} \hookrightarrow P_k\mathbf{ST}>$$ // *Inference*-driven behaviors (B_{inf})

}

5. Cognitive Complexity of Software

1. Introduction
2. From classic and modern information theories to cognitive informatics
3. Cognitive informatics and the brain
4. Cognitive informatics for SE
➔ **5. Cognitive complexity of software**
6. Summary
7. Assignment

Orientations to Software Complexities: SE vs. CS

- **Throughput vs. Functional Complexity**

  The complexity theories of computation and software engineering are different. The former is focused on the problems of *high throughput complexity* that are computing *time efficiency* centered; while the latter puts emphases on the problems of *functional complexity* that are human *cognition time* and *workload* oriented.

5.1 The Cognitive Weights of Software

- **Theorem 6.4**

A *program* $\mathfrak{P}$ is a composition of a finite set of *k* processes according to the time-, event-, and interrupt-based process dispatching rules, i.e.:

$$\wp = \mathop{R}_{k=1}^{m} (@e_k\mathbf{S} \hookrightarrow P_k) = \mathop{R}_{k=1}^{m} [@e_k\mathbf{S} \hookrightarrow \mathop{R}_{i=1}^{n-1} (s_i(k)\ r_{ij}(k)\ s_j(k))],\ j = i + 1$$

The Relative Cognitive Weights of BCS's

- The ***basic control structures (BCS's)*** **are a set of essential flow control mechanisms that are used for expressing and building the logical structure of software.**

- **The *relative cognitive weight of a BCS*, $w_{BCS}(i)$, $1 \le i \le 10$, is the relative time or effort spent on comprehending the function and semantics of a BCS against that of the sequential BCS, i.e.:**

$$w_{BCS}(i) = \frac{t_{BCS}(i)}{t_{BCS}(1)}, \quad 1 \le i \le 10$$

The Calibrated Cognitive Weights of BCS's

BCS (i)	RTPA Notation	Description	Calibrated cognitive weight (w_i)
1	$\rightarrow$	Sequence	1
2	|	Branch	2
3	|…|	Switch	3
4	R_i	For-loop	7
5	R+	Repeat-loop	7
6	R*	While-loop	8
7	$\rightarrowtail$	Function call	7
8	↻	Recursion	11
9	|| or ∯	Parallel	15
10	↯	Interrupt	22

5.2 The Cognitive Complexities of Software Systems

- The ***operational complexity*** of a software system ***S***, $C_{op}(S)$, is determined by the sum of the cognitive weights of all relational operations ***w***(***k***, ***BCS***) in each components of the system, i.e.:

$$C_{op}(S) = \sum_{k=1}^{n_C} C_{op}(C_k) = \sum_{k=1}^{n_C} \sum_{i=1}^{\#(C_s(C_k))} w(k, i) \quad [\mathrm{F}]$$

- **Theorem 6.5** The symbolic complexity $C_s(S)$ is ***a special case of the operational complexity*** $C_{op}(S)$, where the cognitive weights of all kinds of BCS's, w_i(BCS), are simplified as always one, i.e., $w(k, i) \equiv 1$.

The Architectural Complexity

The ***architectural complexity*** **of a software system** ***S***, $C_a(S)$, **is determined by the number of data objects at system and component levels, i.e.:**

$$C_a(S) = \mathrm{OBJ}(S)) = \sum_{k=1}^{n_{CLM}} \mathrm{OBJ}(CLM_k) + \sum_{k=1}^{n_C} \mathrm{OBJ}(C_k) \quad [\mathrm{O}]$$

where ***OBJ*** **is a function that counts the number of data objects in a given CLM (number of global variables) or components (number of local variables).**

The Cognitive Functional Size of Software

- **Theorem 6.6**

The *cognitive functional size* (CFS) of a software system *S*, *CFS*(*S*), is a product of the operational complexity $C_{op}(S)$ and the architectural complexity $C_a(S)$ and, i.e.:

$$
\begin{aligned}
CFS(S) &= C_{op}(S) \bullet C_a(S) \\
&= \{\sum_{k=1}^{n_C} \sum_{i=1}^{\#(C_s(C_k))} w(k,i)\} \bullet \\
&\quad \{\sum_{k=1}^{n_{CLM}} \mathrm{OBJ}(CLM_k) + \sum_{k=1}^{n_C} \mathrm{OBJ}(C_k)\} \quad [\mathrm{FO}]
\end{aligned}
$$

The Unit of CFS of Software

- Software ***cognitive functional size*** (CFS) is proportional to both its ***operational*** and ***architectural*** complexities.

  - The more the architectural data objects and the higher the operational complicity onto these objects, the larger the functional size of the system.

- **The *unit* of CFS of software systems is a single sequential operation (F) onto an individual data object (O) called one *function-object unit* (FO).**

Case Study 1 – IBS (Implementation A)

IBS_AlgorithmST ({I:: AN, BN}; {O:: ⓢIBSResultBL, IBSumN}) ≜

{ // Specification (a)

MaxN := 65535

→ (� (0 < AN < maxN) ∧ (0 < BN < maxN) ∧ (AN < BN)

→ IBSumN := ((BN - 1) * BN) / 2) - (AN * (AN + 1) / 2)

→ ⓢIBSResultBL := T

| � ~

→ ⓢIBSResultBL := F

→! (@'AN and/or BN out of range, or AN ≥ BN')

)

}

Case Study 2 – IBS (Implementation B)

IBS_Algorithm ({I:: A**N**, B**N**}; {O:: ⓈIBSResult**BL**, IBSum**N**}) $\triangleq$

{ // Specification (b)

Max**N** := 65535

→ (� (0 < A**N** < max**N**) ∧ (0 < B**N** < max**N**) ∧ (A**N** < B**N**)

→ IBSum**N** := 0

→ IBSum**N** := $\underset{i\mathbf{N}=A\mathbf{N}+1}{\overset{B\mathbf{N}-1}{R}} (IBSum\mathbf{N} + i\mathbf{N})$

→ ⓈIBSResult**BL** := **T**

| � ~

→ ⓈIBSResult**BL** := **F**

→! (@'A**N** and/or B**N** out of range, or A**N** ≥ B**N**')

)

}

Case Study 3 – MaxFinder

MaxFinder ({I:: X[0]**N**, X[1]**N**, …, X[n-1]**N** }; {O:: max**N** }) ≜
{
Xmax**N** := 0
→ $\mathop{R}\limits_{i\mathbf{N}=0}^{n\mathbf{N}-1}$ (
� X[i **N**]**N** > Xmax**N**
→ Xmax**N** := X[i **N**]**N**
)
→ max**N** := Xmax**N**
}

Case Study 4 – SIS Sort

SIS**ST**({I:: X[i**N**] **Array** }; {O:: X[s_i**N**] **Array**, ⓢSISResult**BL**) $\triangleq$

{ // <Input:: X[i**N**] : Array | $0 \le$ i**N** $\le$ n**N** -1, $0 \le$ X[i**N**]**N** $\le$ m**N**-1, m**N** > n**N**>

// <Output:: X[s_i**N**] : Array | $0 \le s_i$**N** $\le$ n**N** -1, $x_{s0} \le x_{s1} \le, \ldots, \le x_{si} \le, \ldots, \le x_{sn-1}$ >

// <CLM:: SS[j**N**] : Array | $0 \le$ j**N** $\le$ m**N** –1, $0 \le$ m**N** $\le$ max**N**>

// Initialization

$\underset{j=0}{\overset{m-1}{R}}$ SS[j**N**]**N** := 0

// Self-index sorting

$\underset{i=0}{\overset{n-1}{R}}$ ($\uparrow$ (SS[X[i**N**]**N**]**N**)

// Compression

$\rightarrow$ i**N** := 0

$\underset{j=0}{\overset{m-1}{R}}$ $\underset{\ge 0}{\overset{SS[j]<=0}{R}}$ (X[i**N**] **N** := j**N**

$\rightarrow \downarrow$(SS[j**N**]**N**)

$\rightarrow \uparrow$(i**N**)

)

)

$\rightarrow$ ⓢSISResult**BL** := **T**

}

Case Studies on Software System Complexities

System	Time complexity (C_t [OP])	Cyclomatic complexity (C_m [-])	Symbolic complexity (C_s [LOC])	Operational complexity (C_{op} [F])	Architectural complexity (C_a [O])	Functional complexity (C_a [FO])
IBS (a)	ε	1	7	13	5	65
IBS (b)	O(n)	2	8	34	5	170
MaxFinder	O(n)	3	5	115	5	575
SIS_Sort	O(m+n)	5	8	163	11	1,793

Findings based on the Case Studies

- The *first three measurements* cannot actually reflect the real complexity of software systems in design, representation, cognition, and comprehension.

- Although four example systems are with similar symbolic complexities, their *operational and functional complexities* are greatly different.

  - The symbolic complexity cannot be used to represent the operational or functional complexity of software systems.

- *Symbolic complexity* does not represent the throughput or the input size of problems.

- *Time complexity* does not work well for a system there is no loops and dominated operations, because in theory that all statements in linear structures are treated as zero no matter how long they are. In addition, time complexity can not distinguish the real complexities of systems with the same asymptotic function, such as in Case 2 (IBS (b)) and Case 3 (Maxfinder).

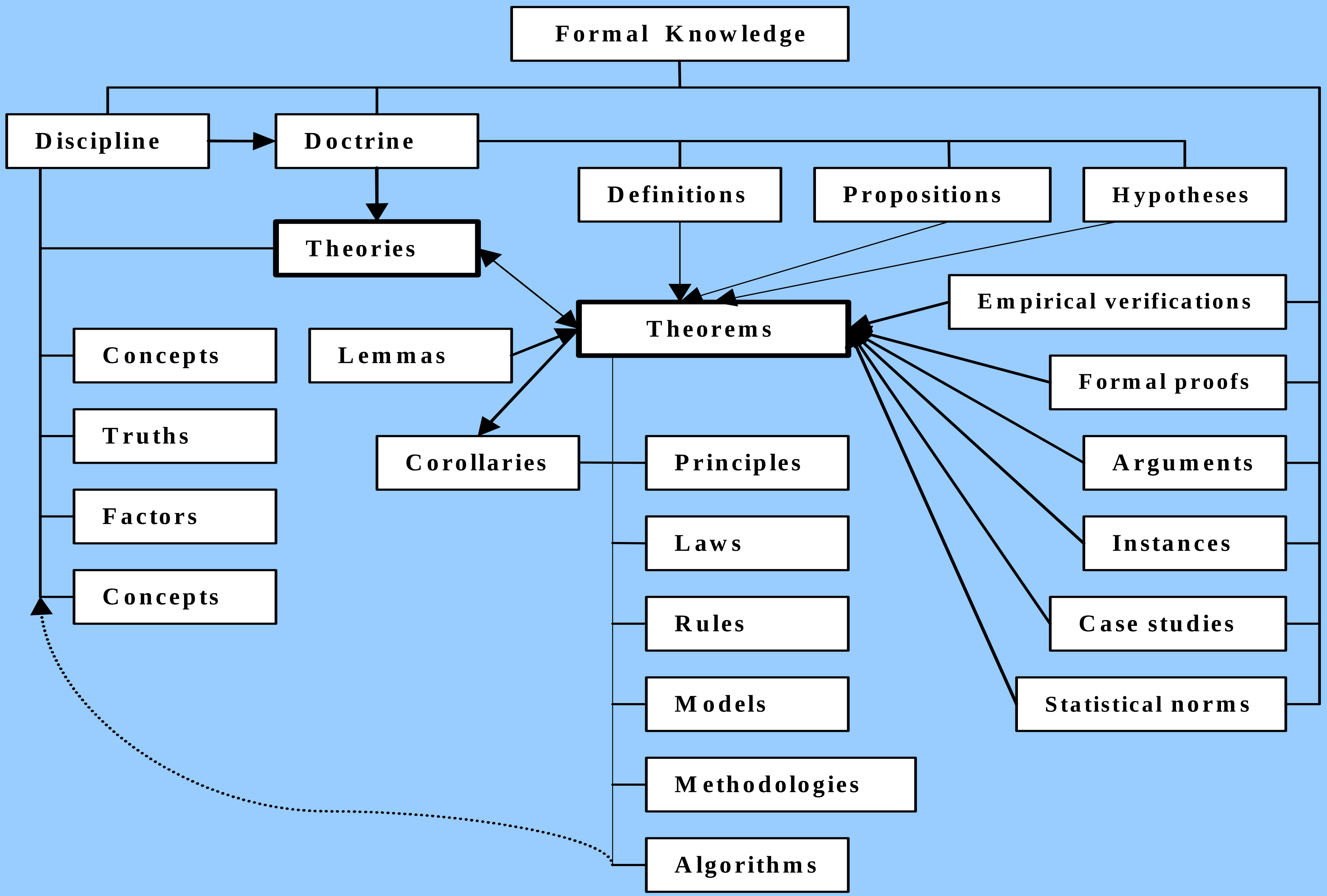
6. Summary

Summary of this Lecture

Lecture 6. Cognitive Informatics Foundations of SE

- **Overview**
  - The brain and natural intelligence
  - Cognitive computing
- **Fundamentals of Informatics**
  - The universal world (IME) model
  - Classical information theory
  - Contemporary information theory
  - Measurements of information
- **Cognitive Informatics and the Brain**
  - Framework of CI
  - The generic αI model (GAIM)
  - The layered reference model of the brain (LRMB)
  - The OAR model for internal knowledge representation
- **Cognitive Informatics for SE**
  - The hierarchical abstraction model (HAM)
  - Informatics properties of software
  - SE psychology
- **Cognitive Complexity of Software**
  - Orientation of software complexities in SE and CS
  - Cognitive weights of software
  - Cognitive complexity of software

The Formalized Knowledge of SE



Three Levels of Research

- Individual problems
- A class of problems
- The whole structure of a class of problems
  - Philosophy
  - Fundamental theories

Research Taste - The Ability to Appreciate

- *Vision* is the art of seeing the invisible.

Method \ Problem	New	Old
New	✓	✓
Old	✓	x

7. Assignment

1. Introduction
2. From classic and modern information theories to cognitive informatics
3. Cognitive informatics and the brain
4. Cognitive informatics laws for SE
5. Cognitive complexity of software
6. Summary

➔ **7. Assignment**

Requirements for the Final Project Report

Following the outline below to develop a final project report for SENG609.19:

1. Summarize the structure of knowledge on the multidisciplinary theoretical foundations of SE delivered in this course in a hierarchical diagram, and describe their interrelationship and possible impacts on SE.
2. You identified a couple of additional fundamental areas for SE in Ex.1.1(b). Try to further describe one of your proposed aspects on the foundations of SE with supporting materials.
3. What are the common characteristics of the pioneers in SE as you reviewed in the classic articles and your group's presentations?
4. Discuss what you gained in this course and where you may use them in your SE practices.
5. Your feedback on the course:
   - (a) Has this course met your expectation?
   - (b) Is this course too easy, just OK, or too hard for you?
   - (c) Are the classic papers helped you to understand the history & major problems in SE?
   - (d) What is your opinion on the overall quality of this course?
   - (e) Provide any comment or suggestion for improving the course in the future.

Requirement: The report is expected longer than 10 pages. An electronic version is required by *Monday, Dec. 14, 2009*.