From Databases to Knowledge-Bases
Kappa , *Quixote* , *Helios*

Kazumasa Yokota

Institute for New Generation Computer Technology (ICOT)
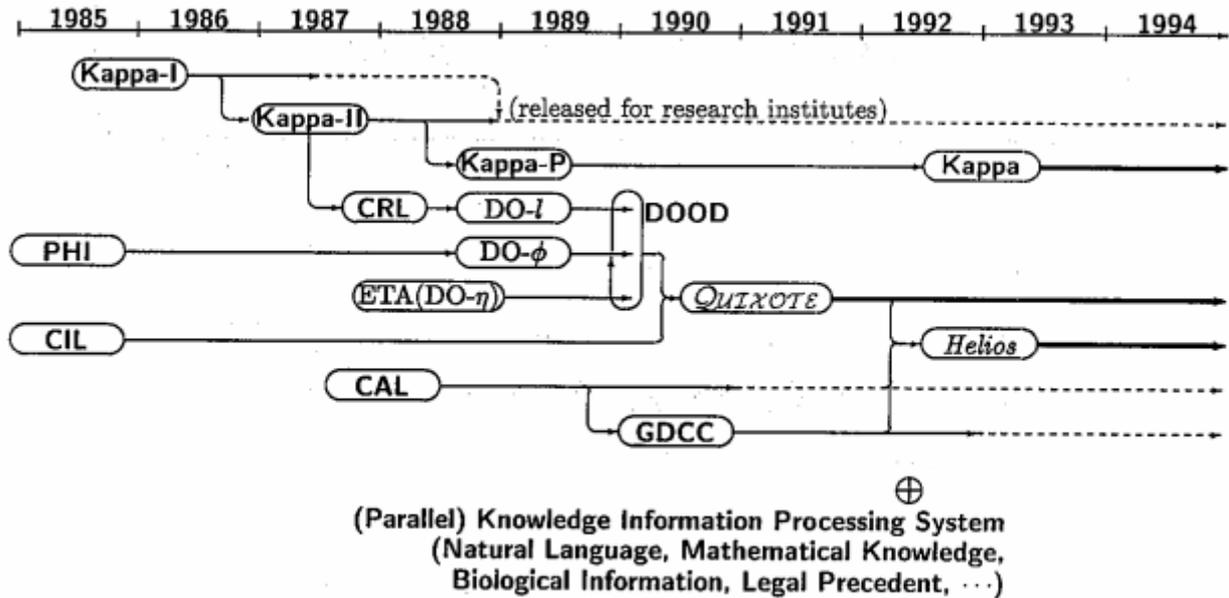
Contents

FGCS

Brief History of R&D on DB and KB

| 1985 | 1986 | 1987 | 1988 | 1989 | 1990 | 1991 | 1992 | 1993 | 1994 |

Kappa-I

Kappa-II → (released for research institutes)

Kappa-P → Kappa

CRL → DO-l → DOOD

PHI → DO-ϕ

ETA(DO-η) → $\mathcal{QUIXOTE}$

CIL → $\mathcal{Helios}$

CAL

GDCC

$\oplus$

(Parallel) Knowledge Information Processing System
(Natural Language, Mathematical Knowledge,
Biological Information, Legal Precedent, ···)

Viewpoint of Database Systems

Helios
> **Extended Multidatabase**
> (including Database Management Systems,
> Constraint Solvers, Knowledge-Base Systems,
> Application Programs, ···)

$\mathcal{QUIXOTE}$
> **Deductive Object-Oriented Database**
> (Extensions of Deductive Databases with
> Object-Orientation Concepts
> Based on Subsumption Constraints)

Kappa
> **Nested Relational Database**
> (Underlying Database Engine)
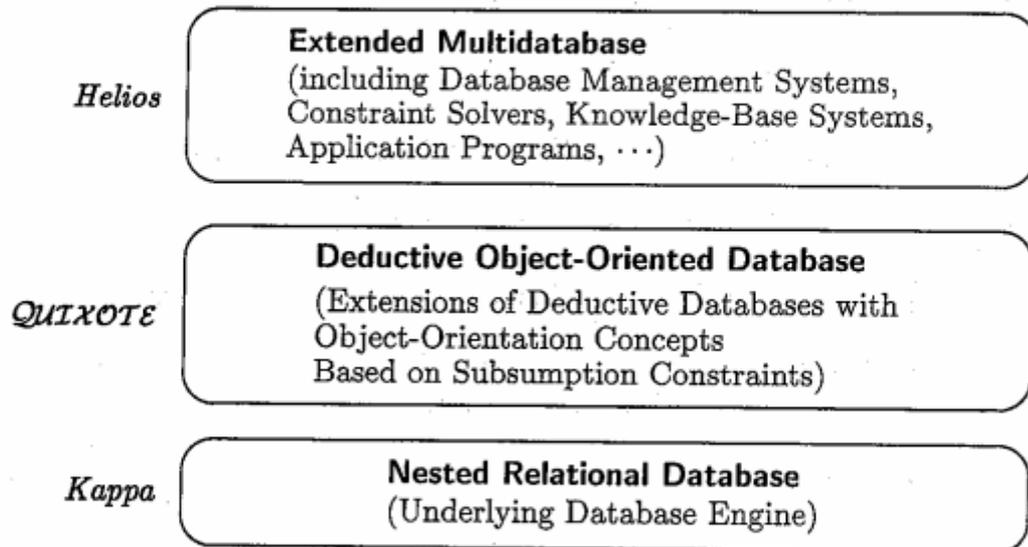
Viewpoint of Knowledge Representation

Helios
> **Heterogeneous Cooperative Problem Solver**
> (Distributed Problem Solving System,
> Multi-Agent Based Systems,
> Multidatabase Systems, ···)

QUIXOTE
> **Knowledge Representation Language**
> (Subsumption Constraints, Rules, Modules,
> Extended Query Processing, ···)

Kappa
> **Underlying Persistent Storage**
> (Hidden from Users)

What is *Kappa* ?

Mass Data

Natural Language Databases
(dictionaries, corpus, ···)
Biological Databases,
Legal Databases,
  ⋮

Applications

Natural Language Processing
Genetic Information Processing
Legal Reasoning
  ⋮

- Nested Relational Model
- Efficient Implementation
- Parallel Processing

Kappa and *CRL*

CRL : Moduralized Deductive Databases Based on the Semantics of *Kappa*

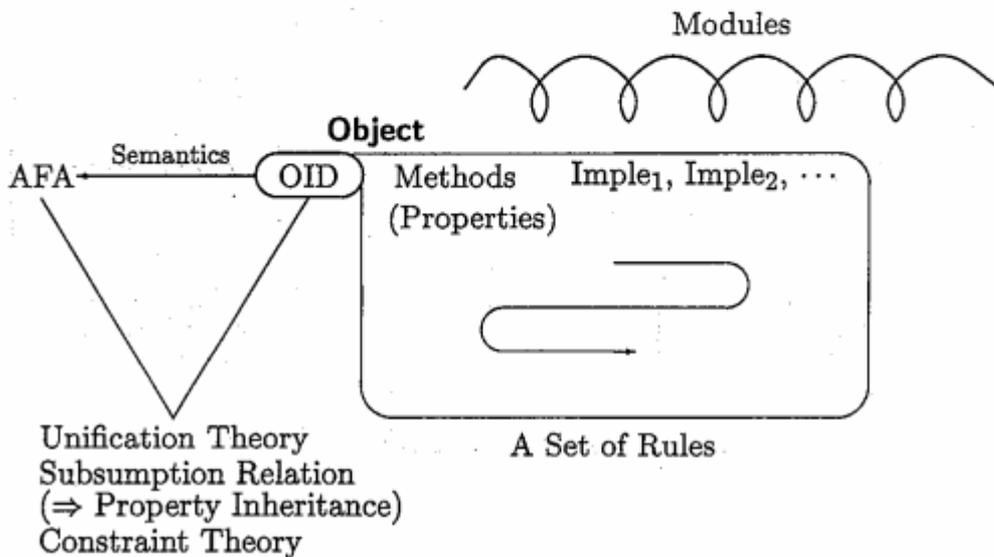$$q \leftarrow p_1, \cdots, p_n$$

$$\equiv q \leftarrow p_1, \cdots, p_{i-1}, p_{i1}, \cdots, p_{im}, p_{i+1}, \cdots, p_n$$

$$\equiv \begin{cases} q_1 \leftarrow p_1, \cdots, p_n \\ \quad \vdots \\ q_k \leftarrow p_1, \cdots, p_n \end{cases}$$

$$\text{where} \quad \begin{aligned} \mathbf{unnest}(p_1) &= \{p_{i1}, \cdots, p_{im}\} \\ \mathbf{unnest}(q) &= \{q_1, \cdots, q_k\} \end{aligned}$$

- Expression-Oriented Semantics
- Set Unification = Constraint Based Set Union
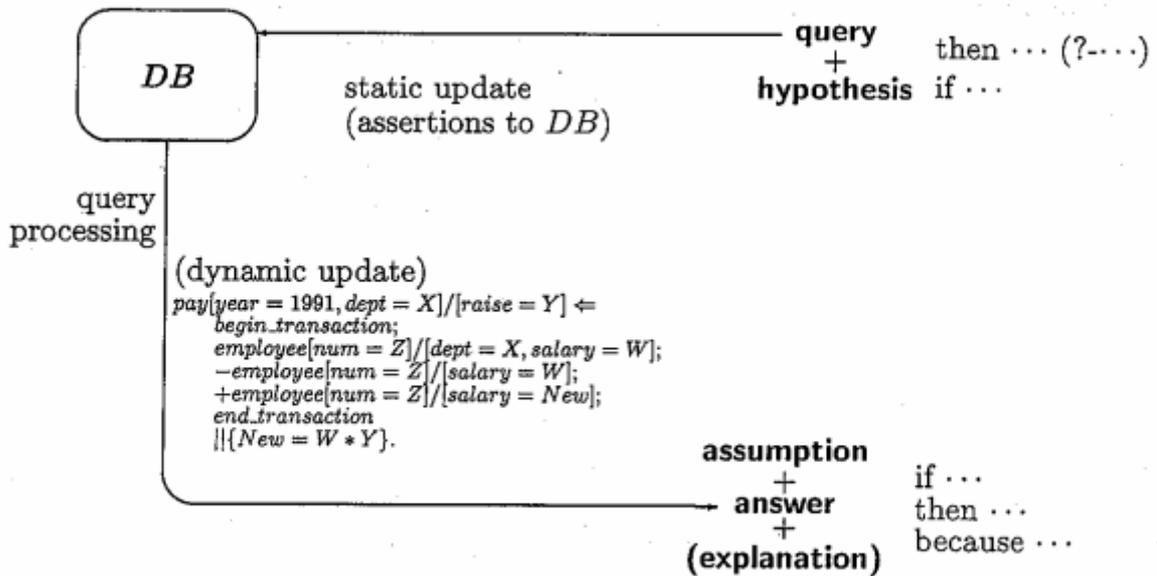- Residue Goals = Bind Hook Mechanism

What is $\mathcal{Q}\textit{uixote}$? (1)

—————— **Logic Programming Language** ——————

What is $\mathcal{Q}\mathit{UIXOTE}$? (2)

$$DB$$

query
+
hypothesis

then $\cdots$ ($?\text{-}\cdots$)
if $\cdots$

static update
(assertions to DB)

query
processing

(dynamic update)
$pay[year = 1991, dept = X]/[raise = Y] \Leftarrow$
  $begin\_transaction;$
  $employee[num = Z]/[dept = X, salary = W];$
  $-employee[num = Z]/[salary = W];$
  $+employee[num = Z]/[salary = New];$
  $end\_transaction$
  $||\{New = W * Y\}.$

assumption
+
answer
+
(explanation)

if $\cdots$
then $\cdots$
because $\cdots$

Example in $\mathcal{Q}\mathit{UIXOTE}$ (Bizet and Verdi)

If Bizet and Verdi are compatriots,
then Bizet is Italian

$bizet/[nat = italian]$
$\Leftarrow compatriot[p_1 = verdi, p_2 = bizet]$

If Bizet and Verdi are compatriots,
then Verdi is French

$verdi/[nat = french]$
$\Leftarrow compatriot[p_1 = verdi, p_2 = bizet]$

$compatriot[p_1 = X, p_2 = Y]$
$\Leftarrow X/[nat = Z], Y/[nat = Z];;$
$bizet;;$
$verdi$

$?\text{-}bizet/[nat = X].$

if $verdi/[nat = italian],$
then $X = italian.$

$?\text{-}verdi/[nat = X].$

if $bizet/[nat = french],$
then $X = french.$
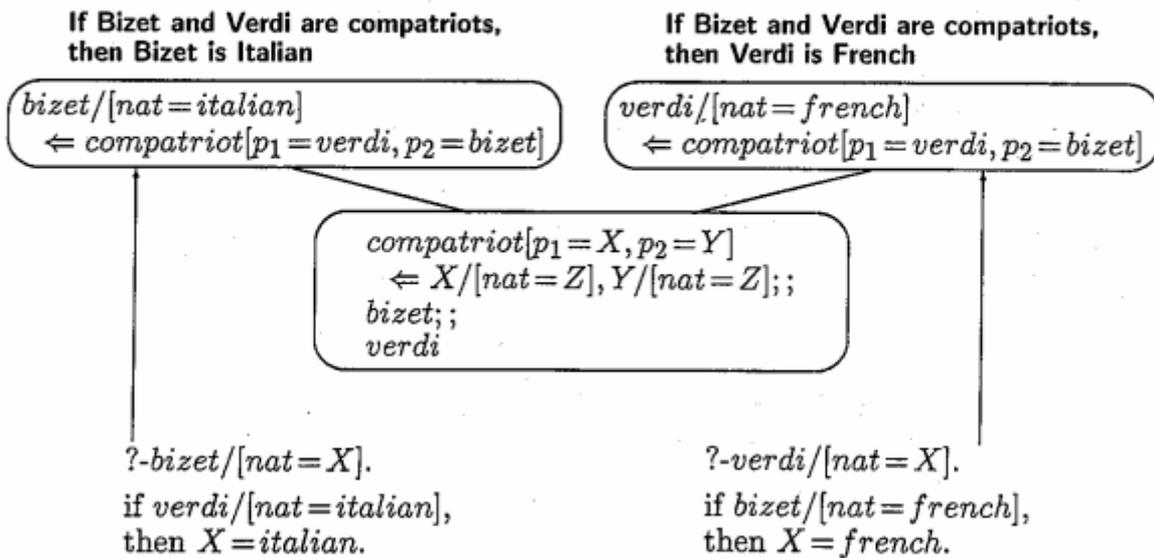
What is *Helios* ? (Motivations)

biological knowledge

- sequence (long string, IR)
- function (rule with structured objects)
- statistics (algebraic constraints)
- ambiguous data (hypothetical data)

legal knowledge

- text (long string, natural language, IR)
- abstracted rule (rule, deduction, abduction)
- inconsistent knowledge (classification)
- negotiation (defendant vs. plaintiff)
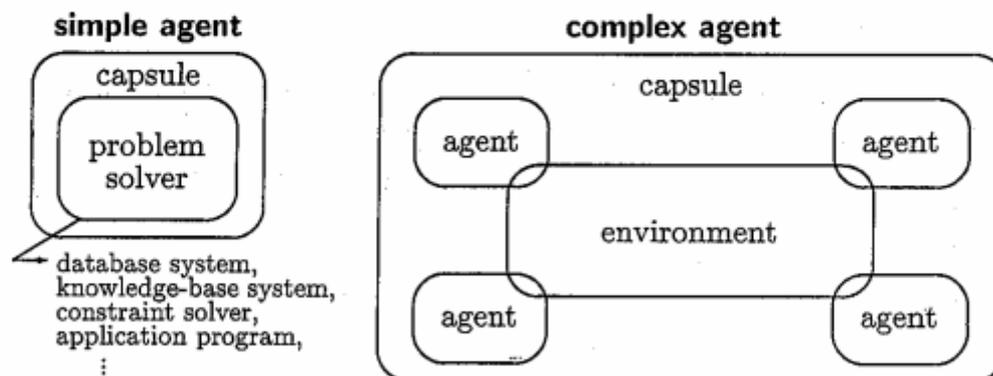- algebraic computation (tax, ⋯)

other applications

- multimedia database
- natural language database
- . . .

$\Longrightarrow$ **heterogeneous cooperative**
databases/constraints/systems

distriburted / parallel processing environment
$\Rightarrow$ **distributed** system

What is *Helios* ? (Testbed)

simple agent

capsule

problem solver

database system,
knowledge-base system,
constraint solver,
application program,
⋮

complex agent

capsule

agent

agent

environment

agent

agent

If an agent detects an unsolvable problem,
the agent throw out the problem into its environment.

User is the outermost environment!

What is *Helios* ? (Languages and Featutres)

████ FGCS ████

agent := (capsule, problem-solver)
| (capsule, environment, $\{agent_1, \cdots, agent_n\}$)

CAPL capsule := (agent-name, methods, self-model,
  negotiation-strategy)

ENVL environment := (agent-names, common-type-system,
  negotiation-protocol, ontology)

Features

- *Message Dispatching*: agent name, method name, function name
- *Negotiation Protocol*: $\dfrac{\text{negotiation process} \cong \text{transaction}}{\text{negotiation protocol} \cong \text{transaction-based protocol}}$
- *Proxy Agent*: efficient negotiation
- *Concurrency Control*: homogenizing its control depending on each solvers
- *Role of User*: end user, the outermost environment, a problem solver

Contributions of *Kappa*, *QUIXOTE*, and *Helios*

████ FGCS ████

Helios
- Comprehensive test-bed for heterogeneous distributed cooperative problem solving
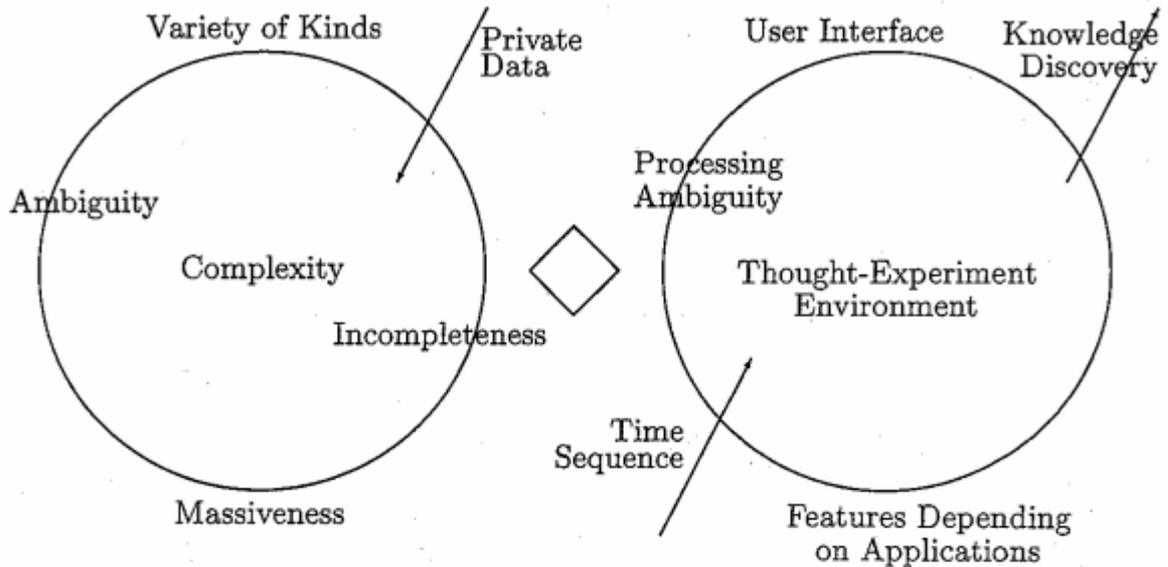- Transaction-based protocol for various kinds of negotiations

QUIXOTE
- Deductive object-oriented database language based on subsumption constraints
- Efficiency and effectiveness for many knowledge information applications

Kappa
- Efficient (parallel) implementation of a nested relational model
- Efficiency and effectiveness as an underlying database layer in data-intensive applications
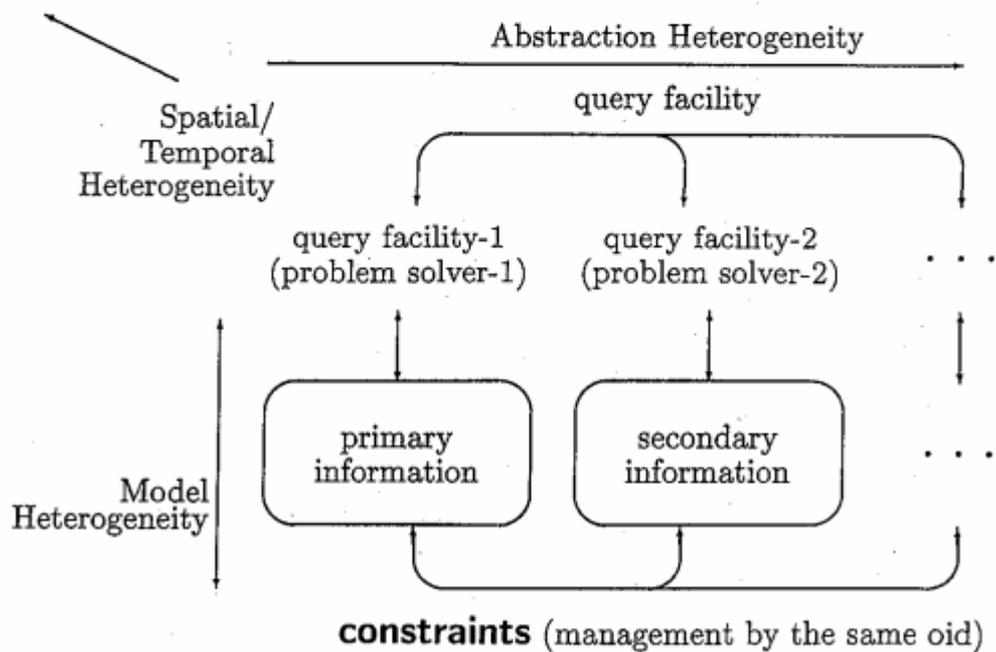
What is Expected of Next Generation DB System? (1)

Variety of Kinds — Private Data — User Interface — Knowledge Discovery — Ambiguity — Processing Ambiguity — Complexity — Thought-Experiment Environment — Incompleteness — Time Sequence — Massiveness — Features Depending on Applications

What is Expected of Next Generation DB System? (2)

Abstraction Heterogeneity — query facility — Spatial/ Temporal Heterogeneity — query facility-1 (problem solver-1) — query facility-2 (problem solver-2) · · · — Model Heterogeneity — primary information — secondary information · · · — **constraints** (management by the same oid)

Concluding Remarks

- *Kappa*, *Quixote*, and *Helios* have been considered under one perspective from database and knowledge-base points of view.

- They can be considered as a step towards next generation database and knowledge-base systems.

- The better applications, the better database and knowledge-base systems. Cooperation from an applications' side is indispensable.

- A ten-year is too short on their R&D activities as as a whole system. They should be continued further for a practical flexible system.

- *Kappa*, *Quixote*, and *Helios* have been released as *ICOT Free Software*. Please enjoy them.