JIIS, 5, 101–112 (1996)

Glossary: Intelligent Integration of Information

GIO WIEDERHOLD gio@cs.stanford.edu
Department of Computer Science, Stanford Univerity, Stanford 94305

This glossary and its base vocabulary were initially established during the Intelligent Integration of Information (I3) Architecture Meeting in Boulder CO, in November 1994, sponsored by ARPA and organized by Roger King and Richard Hull of the University of Colorado. It was subsequently refined during an I3 architecture meeting held in January 1995, organized by Mike Genesereth of Stanford University. It has received inputs from many members of the community, although closure on the I3 architecture specification itself has not yet been achieved. The architecture document is currently being maintained at George Mason University, as ⟨http://isse.gmu.edu/I3_Arch/index.html⟩. Related material on I3 technology can be found in files of the Stanford Logic Group ⟨http://logic.stanford.edu/...⟩.

This glossary is also available on-line, arranged logically and clickable for rapid perusal ⟨http://db.stanford.edu/pub/gio/1994/vocabulary.html⟩. This printed version, because it could not be made active, has the terms in each section arranged alphabetically. Project references and other ancillary material, available on-line, have also been omitted here, since they are quite transient. Please email corrections or additions to ⟨gio@cs.stanford.edu⟩.

This glossary is structured into five logical sections:

1 Architecture
2 Services
3 Resources
4 Ontologies
5 Related Terms

Terms in *italics* are defined within this glossary; when such terms adjoin, they are separated by a | symbol.

1. Architecture

Architecture ⊨ arrangement of *components*.
The following terms define architectural concepts for I3.

actionable information ⊨ *information* which causes the *customer* to initiate *events*.

agent ⊨ *tool* which performs a *service*, either for its *owner* or for a *customer* of its *owner* [GeneserethK:94] [GeneserethSS:94].

application ⊨ a persistent or transient *configuration* of *components*, *glue*, and human-to-computer interaction)*HCI*) modules, to solve a *customer* problem; may cover multiple *domains*.

component $\models$ major building block for an *application* | *configuration*. Incorporates *tools* and possibly *domain-specific* | *knowledge*.

configuration $\models$ an instance of an *architecture* for a *customer* | *application*.

constraint rules $\models$ *rules* defining dependencies, including existence, among *objects*; in an *architecture* can define *layer* assignments for *components*, protocols, etc.

content $\models$ *information* result passed on from a *resource*.

coordination services $\models$ covers *resource* | *discovery*, *distribution*, invocation, *scheduling*, etc. [Tourtier:94].

customer $\models$ owner of the *querying* | *application* program, end-user, or client, who derives *value* from the *services* through receipt of *actionable* | *information*.

data $\models$ recorded facts, typically maintainable by clerks.

domain $\models$ topic area, characterized by internally consistent *semantics* and having recognized experts, for instance: finance, electronic parts, vehicle control, diabetes clinic records. Terms and *ontologies* are often ambigious among distinct *domains* [GohMS:94].

facilitator $\models$ *component* which provides *coordination* | *services*, as *routing* | *queries* for a *customer*. A facilitator may create a *configuration*. By definition, facilitators are required to accept runtime submissions of

1 *metadata* and *metainformation*, i.e., *ontologies*, about *resources*
2 logical statements relating disparate concepts (usually definitions)
3 representation and layout format descriptions.

Facilitators are expected to use this encoded *knowledge* to automate the locating, *accessing*, converting, *translating*, or *routing* of *information*. For example, when a new *resource* becomes available, a facilitator will expect a machine-processable description of the new resource, which must be coherent with the existing description used by the facilitator. Then the facilitator can integrate the new resource, or elide an existing one, and the *consumer* can be immediately served with the new resource [GenesereththS:93].

glue $\models$ Software or *rules* needed to link *components* or to *interoperate* among distinct *domains*.

information $\models$ *data* useful to a *customer* [ShannonW:48].

interoperation $\models$ *customer* | *services* effectively combining multiple *resources* and *domains* [W:94]; requires *interoperability*.

knowledge $\models$ *metadata*, *relationships* among terms, *constraint rules*, processing paradigms, . . . , needed to transform *data* into *information*.

layers $\models$ conceptual, gross categorization of *components* and *tools* in a *configuration*. The I3 *architecture* distinguishes three such layers, each providing a distinct category of *services*: (1) *customer* | *applications*, (2) *coordination* and *mediation*, and, (3) *access* to base *resources*, which may require *wrapping*.

level $\models$ a conceptual categorization, where *objects* at a lower level depend on their *ancestors* at a higher level, as defined by the structure of an *ontology* or other *metadata*.

mediation services $\models$ covers *value-added* processing on *query* text and resulting *content*, as query *reformulation* and *distribution*; and *filtering*, *integrating*, and *processing* the *content* of the resulting data, generating new, denser, or more relevant *information*.

mediator $\models$ *component* to provide *mediation* | *services* in response to a *customer*. The management of a mediator, its programs and its *knowledge-base* is the responsibility of a human *owner*. 'A mediator is a software module that exploits encoded knowledge about certain sets or subsets of data to create information for a higher layer of applications.' .. 'It should be small and simple, so that it can be maintained by one expert or, at most, a small and coherent group of experts' [W:92c]. The mediator, as directed by its *owner*, tries to

1 assure stable delivery of *services*, even when *resources* change

2 assess *heterogenity* of *resources* and maintains *tools* to resolve them

3 *integrate* | *information* from multiple *resources*.

4 forward *information* matching the *consumers* | *model*.

The *owners* of mediators take on the effort and authority for managing the information being produced. They may be reimbursed for the *value-added* information *services* being provided. They should be responsive to changes in needs, environments, and resources to remain competitive [WEA:90].

metadata $\models$ descriptive *information* about *data* in a *resource*, includes *domain* assignment, *ownership*, *access* restrictions, *database* | *model*.

metainformation $\models$ descriptive *information* about *services*, base *information*, and *resources*, as *owner*, *ontology* used, representation, capabilities, cost, etc.

metaknowledge $\models$ descriptive *information* about the *knowledge* in a *resource*, can include the description of the *ontology*, its *domain* coverage, *ownership*, *access* restrictions, representation [NechesFFGPS:91].

model $\models$ formal description of capability of *resource* or needs of *customer*.

query $\models$ request by *customer* for named *information*, produces *contents*.

reference architecture $\models$ a guideline and *constraint rules* for an *architecture*.

resource $\models$ accessible *database*, *object server*, *simulation*, *knowledge-base*, etc., provided by a *server*.

service $\models$ function provided by a *tool* in a *component* for a *customer*, directly or indirectly.

text $\models$ *data*, *information*, or *knowledge* in relatively unformatted or semi-formal character-based representation.

tools $\models$ software programs that implements *services*, typically *domain-independent*.

wrapper $\models$ *tool* to *access* known *resources* and *translate* their *objects* [FranchittiKB:94].

wrapping services $\models$ covers the use of *wrapper* and simliar *tools* which adapt interfaces of *legacy* | *resources* to conform with *access* standards and conventions used in *mediation* and *coordination*.

2. Services

Service $\models$ conceptual functionality provided directly or indirectly, for a *customer* by one or more *components*. No specific implementation is implied.

We focus here on services visible to the *customer*. Alternative *configurations* may package *services* differently and render them less visible.

abstraction $\models$ service to reduce *content* volume by bringing *information* to a higher *level* [W:92a].

access $\models$ locating, retrieval or update of *objects* in *resources* for *query*, *update*, or analysis [PastorMF:92].

activeness $\models$ the ability to trigger actions due to an update *event*, as notify a *subscriber* of *events* or cause secondary updates [SeligmanK:93].

added-*value* service $\models$ provides *information* which is more relevant or easier to assimilate, for a cost which is potentially chargeable [W:95l].

advertising $\models$ *resource* or *mediator* | *model* presentation to a *component* or *customer*.

association $\models$ inclusion of additional *information* not explicitly requested by the *customer* but assumed to be relevant due to its *relationships*.

brokering $\models$ a *coordination* | *service* to find the best *resources*.

browsing $\models$ *services* to support *customer* driven *resource* | *discovery*.

caching $\models$ holding *information* at intermediate *level* to improve performance or resolve *temporal mismatches*.

concurrency control $\models$ assuring synchrony of *updates* to *resources*, typically assigned to a *transaction management* system.

configuration tool $\models$ program used in *coordination* to help select and arrange *components* into a specific instance of an *architectural* | *configuration*.

content $\models$ results produced by *resource* in response to *queries*, often voluminous.

content analysis $\models$ processing of text *objects* to create *information*, as *abstracts* or *indexes* [W:93].

description services $\models$ meta *services* that inform *customers* about *services*, *resources* etc., as *explanation*, *resource discovery*, *impact assessment*.

directory $\models$ *service* to locate and assess available *resources* and *services*, as *white pages*, *yellow pages*, etc.

resource discovery $\models$ *service* searching for *resources*, complements *advertising*.

estimator $\models$ low-*level* | *service* which predicts *cost* and/or performance based on *model*, statistics, or sampling.

explanation $\models$ a *mediation* | *service* to present *models* and *metadata* of *resources* and processes used or available to *customer*; may invoke *impact assessment*.

filtering $\models$ a *mediation* | *services* to improve relevance of *query* | *content* | *information*.

fusion $\models$ *integration* as commonly applied to sensor data and images.

impact assessment $\models$ *services* which reports which *resources* will be affected by *queries* or *updates*.

indexing $\models$ creation of lists to *objects* (an index) to speed-up *access* | *services*.

integration $\models$ a *mediation* | *service* to combine intermediate *contents* from multiple, often heterogeneous *resources*.

mediator instantiation $\models$ populate a *domain-independent* | *mediation* | *service* | *tool* with *domain-specific* | *knowledge*.

matchmaking $\models$ a *coordination* | *service* relating *subscribers* to *advertisers*.

model management $\models$ a *mediation* | *service* to allow *customers* and *mediator owners* to *update* | *models*.

monitoring $\models$ observing *resource* or *virtual data* and creating a *triggering* action when *state* changes occur.

multi-state services $\models$ create, *access*, monitor, and manipulate *state* of virtual data.

optimization $\models$ rearranging and/or *scheduling* processing of a *query* to reduce *cost* or *reponse*-time.

processing $\models$ in *mediating* | *services* manipulates obtained *contents*, typically to increase *information* density and *relevance*, as *integration*, *filtering*, content analysis, summarization, seeking exceptions versus a *user* | *model*, *abstraction*, etc

query decomposition $\models$ determine subsidiary *queries* to *match* available *services* and/or *resources*.

query reformulation $\models$ program to *optimize* or *relax* | *queries* [FlorescuRV:95].

ranking $\models$ a *mediation* | *service* to assign *values* to *retrieved* | *content* | *objects*.

reasoning $\models$ method used in *component* | *services* to draw logical inferences.

relaxation $\models$ *query reformulation* service to produce a relevant superset *content* for a *query*; may use an *ontology mapping* [ChuMB:93].

routing $\models$ a *coordination* | *service* to locate and invoke *resources* and *mediation* | *service*, create a *configuration* dynamically; will use a *directory*.

scheduling $\models$ a *coordination* | *service* to determine order of invocation of *accesses* and other *services*; optimal scheduling is often based on their *estimated* | *costs* or *prices*.

semantic indexing $\models$ *indexing* using indirect terms via an *ontology*.

spatial match $\models$ a *mediation* | *service* to recognize and resolve differences distance measures and *granularities* in *resource* | *data*.

subscription $\models$ request by a *component* or *customer* to be informed of an *event*, similar to a stored *query*.

summarization $\models$ service to aggregate *data* into higher *level* | *objects*.

temporal match $\models$ a *mediation* | *service* to recognize and resolve *time-stamp* differences in *resource* | *data*.

text processing $\models$ a *mediation* | *service* which operates on *text*, as *search*, *indexing*, *categorization*, *editingf*, spelling correction, *translation*, *domain* adaptation.

transaction service $\models$ *service* to assure temporal consistency of *content*s, performed by *transaction management*.

translation $\models$ transforming data into the form and syntax required by the receiving *consumer* [HammerGIPUW:95].

updating $\models$ transmitting *data* changes to the *resources*, requires *transaction* control. We do not assume that all updates flow through *coordination* and *mediation* | *layers*, but rather that most updates are local to the *resource*s.

white pages $\models$ *directory service* to locate individuals.

yellow pages $\models$ *directory service* to locate businesses.

3. Resources

Resource $\models$ *access*ible *database*, *simulation*, *knowledge base*, . . ., includes *legacy* resources.

Resources enable the implementation of I3 systems.

active database $\models$ *database* which exhibits *activeness*.

ancestor $\models$ *object* at a higher level, source of inheritable attributes.

change *state* $\models$ caused by an *update*, insert, delete action.

cost $\models$ price of providing a *service* or *access*ing an *object*; can include size of *content*.

database $\models$ *resource* comprising a collection of *data* with a descriptive *schema*.

database model $\models$ formalized description of *database* | *resource*, includes *schema*.

deductive database $\models$ *database* able to handle logical *rule*s and process *data* accordingly.

dictionary $\models$ listing of terms, expanded *index*, part of an *ontology*.

event $\models$ reason for change in *state* within a *resource* or *component*.

heterogeneity $\models$ the mismatch found in autonomous *resource*s and *service*s, may be found among computing platforms, operating systems, *database* systems and *models*, data representations, *concepts*, *ontolog*ies, *semantic*s, scope of abstract *objects*, and *processing* paradigms [DeMichiel:89].

hierarchy $\models$ *model* structure which assigns *object*s to *level*s, and defines for each *object* a single *ancestor*.

historical data $\models$ *temporal* arranged *data*. Such data is not *updated* with by changing values; new data is only *time-stamped* and appended.

interoperability $\models$ the capability to *interoperate*; critical at the *transport* | *layer*, depends on *standards* and *mediation* to resolve *heterogeneities*.

knowledge base $\models$ *resource* comprising a collection of machine-processable *knowledge*, often in the form of *rules* and *metadata*; enabling *access*ing and computation over *databases* or other *resources*.

legacy resources $\models$ pre-existing or autonomous *resources*, not designed to *interoperate* within a general and flexible *architecture*.

metadata repository $\models$ *database* containing *metadata* and/or *metainformation*.

network $\models$ *model* structure with arbitrary *relationships* among *objects*

object $\models$ an identified instance or abstraction in a *resource*, *customer* | *model*, or a *tool*; has *values* for its attributes.

object *server* $\models$ provider of *data* | *objects* [QianR:95].

owner $\models$ individual or organization who created or obtained rights to an *object*, /I *component*, or *service*, and may derive income from it.

owner of service $\models$ authorized individual or organization responsible for provided *services*.

price $\models$ charges for a *service*, includes *cost*, overhead, and profit.

restructuring $\models$ rearranging data according to an alternate *hierarchy* or *object* | *model*

root object $\models$ the ultimate *ancestor*.

rule $\models$ a logical statement, unit of processable *knowledge*.

rule management system $\models$ *domain-independent* software to collect, select, and act on *rules*.

schema $\models$ listing of *database* relations, attributes, and, optionally, *classes*, *relationships*, *constraints*, limits, other *metadata*. Basis for a *resource* | *ontology*.

simulation $\models$ *resource* able to generate hypothetical *data* or *information*, often to project *state* into the future, based on a *model*.

state $\models$ an instance or version of a *data* or *information* base.

transaction impact $\models$ reports which *resources* are affected by an *update* to an *impact assessment* | *service*.

transaction management $\models$ assuring that temporal consistency of *databases* is not compromised by *updates*.

value $\models$ *content* metric in *customer* model, as quality, relevance, up-to-dateness, cost.

view $\models$ subset of a *database* | *constrained* to a *domain*, and *restructured* as required [LuMSS:95].

view-object $\models$ *data object* constructed from a *database* | *view* and *model* [BarsalouW:89].

virtual data $\models$ *data* represented by references and procedures.

warehouse $\models$ a *database* containing or providing *access* to selected, *abstracted*, and *integrated* | *data* from multiple *resources*; typically redundant with respect to the source data, unless the sources do not keep *historical* data [ZhouHKF:96].

4. Ontologies

Ontology $\models$ the specification of a conceptualization [Gruber:93], i.e., the set of terms and their *relationships* used in a *domain*, denoting concepts and *objects*.

Ontologies encompasse many earlier notions of descriptive data, for example, this document for I3, *metadata*, | *metainformation*, and *metaknowledge*.

concept $\models$ term which defines an *abstract* | *object* or *aggregation* of *object*s, with its *relationship*s.

class $\models$ defines *metaknowledge* as methods, attributes, *ancestry*, etc. for *object* instances assigned to it.

domain-specific $\models$ scope constrained to a single *domain*, assumes coherence, i.e., absence of *semantic* mismatches.

domain-independent $\models$ software, *tools*, and global *knowledge* which is applicable in multiple *domain*s.

editing $\models$ processing of *text* to assure conformance with an *ontology*.

is-a $\models$ *relationship*s. among *class*es, and hence their *object*s, which inherit attributes from their *ancestor*s

matching rules $\models$ statements to define term equivalence among *domain*s, may induce *uncertainty*.

merged ontology $\models$ *ontology* created by combining several *ontolog*ies; may require *mapping* | *rules*.

ontology algebra $\models$ set of operations to achieve composable *merging*, intersecting, *subsetting*, and *mapping* of *ontolog*ies.

ontology comparator $\models$ *tool* to determine equivalence, subsumption, or intersection of *ontolog*ies; used to determine the *rul*es needed for their *mapping*.

ontology mapping $\models$ transformation of terms among *ontolog*ies; uses *matching rules*, often needed to link *customer*s to *resource*s.

schema transformation $\models$ adaptation of a *schema* to an alternate *ontology*.

relationship $\models$ linkage among terms, as reference, *is-a*, dependency, part-of, subset, derivation, etc.; may be *domain*-specific.

semantic $\models$ pertaining to the meaning of a term, often expressed as a set of *relationship*s.

shared ontology $\models$ subset of *ontolog*ies agreed on by multiple *customer*s.

syntactic $\models$ pertaining to the format or ordering of terms, often expressed as *constraint*s.

temporal consistency $\models$ achieved if all data that are not otherwise *time-stamped*; refer to the same time instance and have the same temporal *granularity*.

uncertainty $\models$ augmentation of numeric or logic values with ranges or probabilities.

5. Related Terms

These terms refer to technologies that are often essential to I3 efforts, but that are not the prime focus of the research area.

actor $\models$ an *agent* that can operate autonomously and set its own goals. The I3 paradigm does not require actor capabilities.

agile manufacturing $\models$ flexible manufacturing *configur*ations [MorrisMDF:94].

authentication $\models$ validation that an identifier belongs to the right *customer*.

authorization $\models$ *access* or processing rights held by a *customer*.

client-*server* architecture $\models$ direct linkage of *customer*s to *resource*s providing *data* or *object*s.

decision-making $\models$ *customer* | *application* assessing and choosing alternatives in *plan*ning, may involve disjoint *domains* [LewisS:93].

distributed problem solving $\models$ *application* involving *interoperation* at the *coordination* | *layer* with remote *resources*.

encryption $\models$ technology to safeguard content from un-authorized *customer*s or invaders.

federated architecture $\models$ multiple *client-server* linkages, with a shared *schema* and *ontology*

granularity $\models$ size of atomic measurements for data, as seconds, . . . , years; or angstroms, . . ., miles.

HCI $\models$ human, i.e., *customer*, to computer interaction, delegated in I3 to the application layer, can be served via a *customer* /I *model* in the *mediator*.

human collaboration $\models$ *interoperation* at the *customer* | *layer*, can be aided by *mediating* | *tools* [McGuireKWTGO:93], [CutkoskyEFGGMTW:93].

hypothetical access $\models$ analysis of *impact* or *cost* of *plan*ned | *access*, based on *model*.

learning $\models$ *updating* of one's *knowledge-base* [MitchellCFDZ:94].

lock manager $\models$ *tool* control and synchronize *access* to a *resource*.

middleware $\models$ collective term for *wrapper*s, *object* | *service*s, *mediator*s, *facilitator*s.

National Information Infrastructure (NII) $\models$ collection of common network *accessible* *services*, *resource*s, *storage* and *transport* mechanisms.

node $\models$ *component*s in an information network *configuration*, associated with a *site*.

planning $\models$ an *application* requiring *access* to a variety of *information* to establish current *state* and *simulation* to assess alternative future *state*s [ArensCHIK:94].

precision $\models$ comprehensiveness of *content*s: (amount retrieved)/(amount available).

privacy $\models$ right to limit *access*; held by subject described in *data* [JonesCW:95].

relevance $\models$ usefulness of *contents*: (amount of interest)/(amount retrieved).

resolution $\models$ *granularity* of *content*.

responsiveness $\models$ adequacy of retrieval speed: 1/(time for first *content*).

security $\models$ assurance that *private* and protected *information* is only provided to *authenticated* and *authorized* | *component*s and *customer*s [GongQ:96].

server $\models$ node in a computer-network which provides *resources* to be used by other nodes, as in a *client-server* or *mediate* | *architecture*s. The role can be transient, a server may be a *customer* in an alternate *configuration*.

site ⊨ source, sink, or forwarding computer in a communication network, denotes a location. *Node*s that share a site exhibit less *heterogeneity*.

standard ⊨ *metadata* protocol to provide *syntactic* | *interoperability*.

time-stamp ⊨ label for an information or data unit that specifies its temporal interval of validity.

transport ⊨ the actual shipping of uninterpreted *data* among *node*s.

type abstraction hierarchy ⊨ multi-*level* | *knowledge* representation for *query* | *relaxation* and *semantic indexing*.

version ⊨ a maintained alternative of an *object*, often needed to maintain consistency with past *application*s or *content*s.

References

We include here references cited in the foreword as well as references cited in the glossary itself.

[ArensCHIK:94] Yigal Arens, Chin Chee, Chun-Nan Hsu, Hoh In, and Craig Knoblock: "Query Processing in an Information Mediator"; *Proc. ARPA/Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop*, Tucson AZ, Morgan Kaufmann, 1994.

[BarsalouW:89] Thierry Barsalou and Gio Wiederhold: "Knowledge-directed Mediation Between Application Objects and Data"; *Proc. Working Conf. on Data and Knowledge Integration*, Un. of Keele, England, 1989, Pittman Pub.

[Catell:94] R.Cattell (ed): *The Object Database Standard: ODMG* (3 9version 1.1); Morgan Kaufman, May 1994.

[ChuMB:93] W.W. Chu, M. Merzbacher, and L. Berkovich: "The Design and Implementation of CoBase"; *ACM SIGMOD '93*, May 1993, pp. 517–522.

[ChuQ:94] W.W. Chu and Q. Chen: "A Structured Approach for Cooperative Query Answering"; *IEEE Transactions on Knowledge and Data Engineering,* Vol.6 No.5, October 1994.

[CourtneyJSSW:93] A. Courtney, W. Janssen, D. Severson, M. Spreitzer, and F. Wymore: *Inter-Language Unification* rel.1.5; Xerox PARC, ISTL-CSA-94-01-01, No.P94–58, May 1994, ftp://ftp. parc.xerox.com/pub/ilu/ilu.html.

[CutkoskyEFGGMTW:93] M. Cutkosky, R. Engelmore, R. Fikes, T. Gruber, M. Genesereth, W. Mark, J. Tenenbaum, and J. Weber: "PACT: An experiment in integrating concurrent engineering systems"; *IEEE Computer*, January 1993.

[DeBellisH:95] Michael DeBellis and Christine Haapala: "User-centric Software Engineering (USE)"; *IEEE Expert*, Feb. 1995, Vol.10, No.1.

[DeMichiel:89] Linda DeMichiel: "Performing Operations over Mismatched Domains"; *IEEE Transactions on Knowledge and Data Engineering*, Vol.1 No.4, Dec. 1989.

[FininFMM:94] Tim Finin, Richard Fritzson, Don McKay, and Robin McEntire: "KQML as an Agent Communication Language"; *Proc. 3rd International Conference on Information and Knowledge Management* (CIKM'94), ACM Press, November 1994;

[FlorescuRV:95] Daniela Florescu, Louiqa Raschid, and Patrick Valduriez: "Using Heterogeneous Equivalences for Query Rewriting in Multidatabase Systems"; *Proc. International Conference on Cooperative Information Systems (CoopIS95)*, Vienna Austria, May 1995, available from ⟨coopis@cs.toronto.edu⟩.

[FranchittiKB:94] J.C. Franchitti, R. King, and O. Boucelma: "A Toolkit to Support Scalable Persistent Base Infrastructures"; in *Proceedings of the Sixth International Workshop on Persistent Object Systems*, Tarascon, France, September 1994, Springer-Verlag LNCS.

[GeneserethS:93] Michael R. Genesereth and Narinder P. Singh: *A Knowledge Sharing Approach to Software Interoperation*; Stanford University, 1993.

[GenesererthK:94] M.R. Genesereth and S. Ketchpel: "Software Agents"; *Comm. of the ACM*, Vol.37 No.7, July 1994, pp. 48–53.

[GeneserethSS:94] Michael R. Genesereth, Narinder P. Singh and Mustafa A. Syed: "A Distributed and Anonymous Knowledge Sharing Approach to Software Interoperation"; *Proc. Int. Symp. on Fifth Generation Comp Systems*, ICOT, Tokyo, Japan, Vol.W3, Dec.1994, pp. 125–139.

[GohMS:94]. Cheng Hian Goh, Stuart Madnick, and Michael Siegel: Context Interchange: Overcoming the Challenges of Large-Scale Interoperable Database Systems in a Dynamic Environment"; in Proc. 3rd Int'l Conf on Information and Knowledge Management (CIKM 94), pp.337–346. Gaithersburg, MD, Nov 1994.

[GongQ:96] L. Gong and X. Qian: "Computational Issues in Secure Interoperation"; to appear in *IEEE Transactions on Software Engineering*, IEEE, January 1996.

[GreenLM:91] Cordell Green, Michael Lowry and Robert McCartney: "Automating Software Design: An Overview"; in *Automating Software Design*, AAAI/MIT Press, 1991.

[Gruber:93] T.R. Gruber: "A Translation Approach to Portable Ontologies'; *Knowledge Acquisition*, Vol.5 No.2, 1993, pp. 199–200.

[HammerGIPUW:95] Joachim Hammer, Hector Garcia-Molina, Kelly Ireland, Yannis Papakonstantinou, Jeffrey Ullman, and Jennifer Widom: "Information Translation, Mediation, and Mosaic-Based Browsing in the TSIMMIS System"; *Proc. ACM SIGMOD International Conference on Management of Data*, San Jose, California, June 1995.

[JonesCW:95] V.E. Jones, N. Ching, and M. Winslett: "Credentials for Privacy and Interoperation"; *Proc. New Security Paradigms '95 Workshop*, La Jolla, California, August 22-25, 1995.

[LewisS:93] M. Lewis and K. Sycara: "Informed Decision Making in Multi-Specialist Cooperation"; *Group Decision and Negotiation*, Vol.2 No.3, October 1993.

[Libicki:95] Martin Libicki: *Information Technology Standards*; Digital Press, 1995.

[LuMSS:95] James J. Lu, Guido Moerkotte, Joachime Schue, and V.S. Subrahmanian: "Efficient Maintenance of Materialized Mediated Views"; *Proc. International Conference on Management of Data (Sigmod 95)*, ACM, 1995.

[MacGregor:90] Robert MacGregor: "The Evolving Technology of Classification-based Knowledge Representation Systems"; in John Sowa, editor, *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, Morgan Kaufmann, 1990.

[MarkTMS:92] William Mark, Sherman Tyler, James McGuire, and Jon Schlossberg: "Commitment-Based Software Development"; IEEE Transactions on Software Engineering, October, 1992.

[McGuireKWTGO:93] James G. McGuire, Daniel R. Kuokka, Jay C. Weber, Jay M. Tenenbaum, Thomas R. Gruber, Gregory R. Olsen: "SHADE: Technology for Knowledge-Based Collaborative Engineering"; *Concurrent Engineering: Research and Applications*, Vol.1 No.3, September 1993.

[Miller:56] George Miller:"The Magical Number Seven ± Two"; *Psych.Review*, Vol.68, 1956, pp.81-97.

[MitchellCFDZ:94] T. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski: "Experience with a Learning Personal Assistant": *Comm. ACM*, Vol.37 No.7, pp. 81–91, July 1994.

[MorrisMDF:94] KC Morris, Mary Mitchell, Chris Dabrowski, and Elizabeth Fong: "Database Management Systems in Engineering"; in the *Encyclopedia of Software Engineering*, pp. 282–308, John Wiley and Sons, 1994.

[Musen:92] M.A. Musen: "Dimensions of Knowledge Sharing and Reuse"; *Computers and Biomedical Research*, Vol.25, pp. 435–467, 1992.

[NavatheD:95] Shamkant B. Navathe and Michael J. Donahoo: "Towards Intelligent Integration of Heterogeneous Information Sources"; *Proc. 6th International Workshop on Database Re-engineering and Interoperability*; March 1995.

[NechesFFGPS:91] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. Swartout: "Enabling Technology for Knowledge Sharing"; *AI Magazine*, Vol.12 No.3, 1991, pp. 36–56.

[OMG:91] Object Management Group: *The Common Object Request Broker: Architecture and Specification*; OMG Document 91.12.1, OMG and X/Open, distributed by QED-Wiley, Wellesley MA, 1991

[PapakonstantinouGU:96] Yannis Papakonstantinou, Hector Garcia-Molina, Jeffrey Ullman: "Medmaker: A Mediation System Based on Declarative Specifications"; *International Conference on Data Engineering*, New Orleans, IEEE February 1996.

[PastorMF:92] J.A. Pastor, D.P. McKay, and T. Finin: "View-Concepts: Knowledge-based Access to Databases"; *Proc. 1st International Conference on Information and Knowledge Management (CIKM92)*, Baltimore, MD. 1992.

[QianR;95] X. Qian and L. Raschid: "Query Interoperation among Object-oriented and Relational Databases"; *Proc. 11th International Conference on Data Engineering*, IEEE pp. 271–278. March 1995.

[Rowe:95] N.C. Rowe: "Retrieving Captioned Pictures Using Statistical Correlations and a Theory of Caption-Picture Co-reference"; *Fourth Annual Symposium on Document Analysis and Retrieval*; Las Vegas NV, April 1995.

[SeligmanK:93] Len Seligman and Larry Kerscherg: "An Active Database Approach to Consistency Management in Data- and Knowledge-based System"; *Journal of Intelligent Information Systems*, Vol.2 No.2, 1993.

[ShannonW:48] Claude E. Shannon and Warren Weaver: "The Mathematical Theory of Computation"; *Bell System Technical Journal*, 1948,

[ShivakumarG:96] N. Shivakumar and H. Garcia-Molina: "Building a Scalable and Accurate Copy Detection Mechanism"; to appear in *Proc. 1st ACM Conference on Digital Libraries (DL'96)*; ACM, March 1996.

[Tourtier:94] Paul-Andre Tourtier: "A Flexible, Facilitator-based Cooperation Framework"; *Proc. Int. Symp. on Fifth Generation Comp Systems*, ICOT, Tokyo, Japan, Vol.W3, Dec.1994, pp. 101–110.

[W:92a] Gio Wiederhold: "The Roles of Artificial Intelligence in Information Systems"; *Journal of Intelligent Information Systems*, Vol.1 No.1, 1992, pp. 35–56.

[W:92c] Gio Wiederhold: *Mediation in the Architecture of Future Information Systems*, *IEEE Computer Magazine*, Vol.25 No.3, March 1992, pp. 38–49; republished in Hurson et al: *Multi-database Systems: an Advanced Solution for Global Information Sharing*; IEEE Press, 1993.

[W:93] Gio Wiederhold: "Intelligent Integration of Information"; *Proc. ACM SIGMOD Conference*, ACM 1993, pp. 434–437.

[W:94] Gio Wiederhold: "Interoperation, Mediation, and Ontologies"; *Proc. Int. Symp. on Fifth Generation Comp. Systems*, ICOT, Tokyo, Japan, Vol.W3, Dec.1994, pp. 33–48.

[W:95l] Gio Wiederhold: "Digital Libraries, Value, and Productivity"; *Com. ACM*, Vol.38 No.4, April 1995, pp. 85–96.

[W:95m] Gio Wiederhold: "Modeling and System Maintenance"; in Michael P. Papazoglou (ed.): *OOER'95: Object-Oriented and Entity Relationship Modelling*; Springer Lecture Notes in Computer Science, Vol. 1021, pp. 1-20, December 1995.

[WEA:90] Gio Wiederhold et al: *A Mediator Architecture for Abstract Data Access*; Stanford University, Report Stan-CS-90-1303, August 1990.

[WG:95] Gio Wiederhold and Michael Genesereth: "Basis for Mediation"; *Proc. International Conference on Cooperative Information Systems (CoopIS95)*, Vienna Austria, May 1995, available from ⟨coopis@cs.toronto.edu⟩, pages 138-155.

[ZhouHKF:96] G. Zhou, R. Hull, R. King, and J.-C. Franhitti: "Supporting Data Integration and Warehousing Using H2O"; to appear, *IEEE Data Engineering Bulletin*, 1996, special issue on Data Warehousing, Jennifer Widom (ed.).