

The 11th European Conference on Knowledge Management

**Universidade Lusíada de Vila Nova de Famalicão,
Famalicão, Portugal**



**The Acquisition and Sharing of Domain Knowledge Contained in Software
with a Compliant SIK Architecture**

by Prof. dr. Vasile AVRAM

**Academy of Economic Studies
Bucharest - Romania**



This slide added after.

The Acquisition and Sharing of Domain Knowledge Contained in Software with a Compliant SIK Architecture

By: Avram, Vasile

Edited by: Tome, E

Conference: 11th European Conference on Knowledge Management Location: Univ Lusiada Vila Nova Famalicao, Famalicao, PORTUGAL Date: SEP 02-03, 2010

PROCEEDINGS OF THE 11TH EUROPEAN CONFERENCE ON KNOWLEDGE MANAGEMENT, VOLS 1 AND 2 Pages: 19-26 Published: 2010

Close Abstract

The benefits and needs for extensive knowledge management towards the organizations require, among other things, to change software architectures for a better adaptation to the knowledge era. The change argued in this paper refers to adding a component in the form of one or many companion knowledge repositories expressed as Resource Description Framework (RDF), to systematically acquire, structure, formalize, store and maintain domain knowledge described as business rules, for all business rules that are incorporated in the software product itself. For every software adopting this architecture, the companion repositories acts as vectors to manage, communicate, transfer and share what is called "software inherited knowledge" (SIK), like a genetic repository. SIK architectures refer mainly to software that incorporates internally, mixed with its logic, the served domain business rules. It refers also partly to software that uses externalized business rules but keeps also business rules internally incorporated and does not refer at all to business rules management systems that define and manipulate externally all business rules. As it is defined, SIK allows applications to inherit domains rules from other applications and direct (or little adapted) usage of those business rules, from the Business Rules Management System (BRMS) the company may have. The inheritance of business rules, by using SIK architecture, is different from the inheritance induced by the internal software classes and objects, in object-oriented applications. One aim of this paper is to define and explain the SIK concept, together with all related concepts, from both theoretical and practical perspectives. Another aim of this paper is to exemplify how software can be designed or adapted to be compliant with SIK architecture. The last aim is to outline and emphasize the major benefits, for all users and environments where the application software is used, when adopting SIK architecture for software applications or when buying applications complying with that architecture.

(from Web of Science Core Collection)



Facts

- ◆ Increasing complexity and dependence of organizations on IT;
- ◆ Competition requires to use both business and IT best practices;
- ◆ Brand software incorporates good practices from those used by leader companies;
- ◆ Good practices are knowledge allowing the application and operation of those practices;
- ◆ An organization adopting and using such software in its current operation, will start its activities and competitiveness at least from that point of good practices and associated knowledge;





Corollary

- ◆ To take advantage of all domain knowledge incorporated in software products the architecture of that must include a component capable to reveal to the user and/or other applications the business domain knowledge incorporated: companion repositories that acts as vectors to manage, communicate, transfer and share what is called “software inherited knowledge” (SIK), like a genetic repository.





Aims of the paper

- ◆ to define and explain the SIK concept, together with all related concepts, from both theoretical and practical perspectives;
- ◆ to exemplify how software can be designed or adapted to be compliant with SIK architecture;
- ◆ to outline and emphasize the major benefits, for all users and environments where the application software is used, when adopting SIK architecture for software applications or when buying applications complying with that architecture.





Aspects of knowledge management when using software

- ◆ software manipulation knowledge (classic sense) -> knowledge about software product and operation of this (acquired in learning, training, and practice-explicit to implicit);
- ◆ - architectural knowledge -> as IT knowledge used to analyze, design, code, etc.
- ◆ - domain knowledge -> domain business rules incorporated in the software product (**mobile similarly to the tacit knowledge of the employees**).



knowledge inherited at product adoption time



Knowledge management and software categories

There are several categories of application software:

- Back-office;
 - Front-office;
 - ERP;
 - eApplications.
- } → -Vertical
 } → -Horizontal
 } → Industry best practices

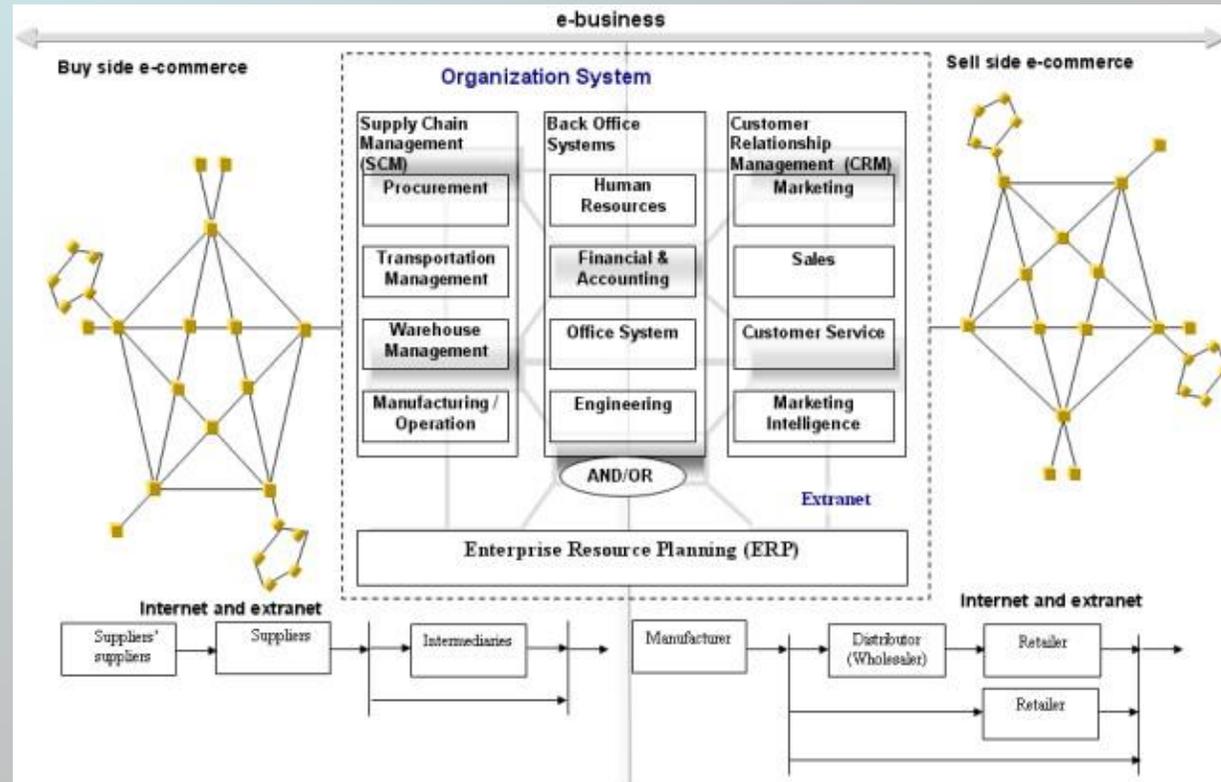


Figure 1: The extended company and management of customers and suppliers relationships (Source: Internet Technologies for Business - Business categories and models in Internet, Vasile Avram, <http://www.avrams.ro>)

Knowledge management and software categories



Software categories depending how incorporate and manipulate modeled domain knowledge:

- BRMS;
- “knowledge externalized”;
- “monolithic”.



SIK Architecture

Two common definition for software architecture:

‘Software architecture presents a view of a software system as components and connectors. Components encapsulate some coherent set of functionality. Connectors realize the runtime interaction between components’ (Albin, 2003: 3);

‘The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them’ (Bass, Clements, and Kazman, 2003: 3)



SIK Architecture

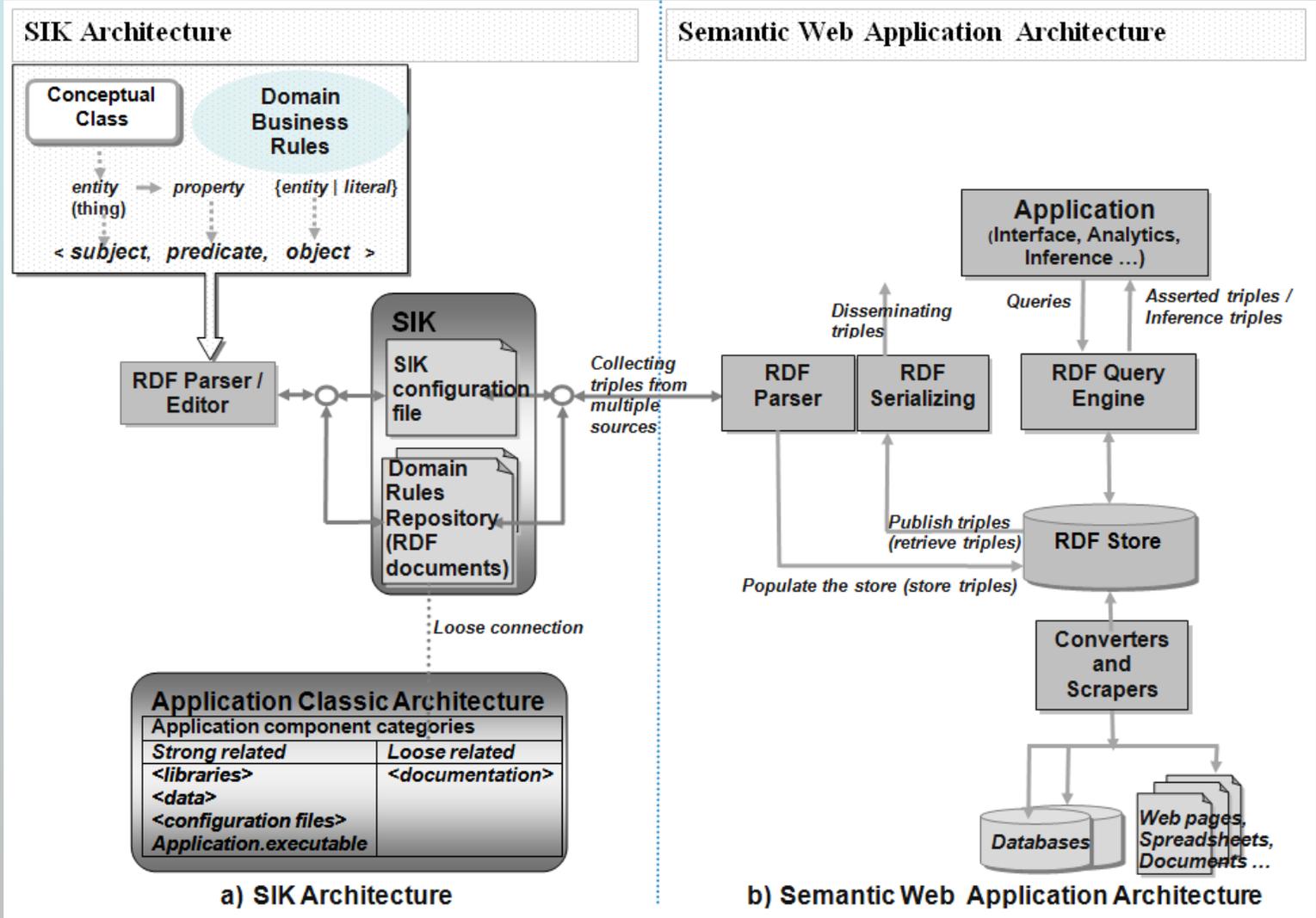


Figure 2: The SIK Architecture in Connection with a Semantic Web Application Architecture

SIK Architecture



SIK (Software Inherited Knowledge) is a component in the form of **one or many knowledge repositories** (as RDF files) to systematically acquire, structure, store and maintain knowledge, formalized as business rules, for all business rules that are incorporated in the software product itself.

SIK:

- Refers to domain knowledge incorporated by software;
- Must be accessible to all persons and/or applications having rights to access and manipulate them;
- Act as a source of enriching every knowledge base deserving / assisting knowledge workers;
- Is a source for preventing the “black-box” syndrome;



SIK Architecture

SIK is an RDF document which in turn is an XML-based language for describing resources. It inherits the characteristics of XML and has his own characteristics, such as:

- **The described resources are well-formed and valid;**
- **Documents and data are application-independent;**
- **Has a standard syntax and structure for metadata and data;**
- **Protection by encryption of the sensitive content;**
- **Mixing encrypted parts with unencrypted ones;**
- **Schema and data are both contained in the same document;**
- **Allows describing multiple independent / dependent business rules in the same physical document;**
- **Is not a new technology.**





Benefits of SIK adoption

By adopting the SIK proposed architecture for a software product, its consumers of this will have at least the following benefits:

- Up-to-the-minute information about the business rules applied through the used software applications;
- Competitive analysis of products or requirements regarding the way to solve;
- Automated reasoner can deduce (infer) conclusions;
- For Web service and SaaS offer the possibility to access directly the business rules described;



Benefits of SIK adoption

- **Competitiveness of the company can be really sustained by knowledge;**
- **Business rules are described in a computer actionable format facilitating application software integration;**
- **Application can be sold as traditional or as knowledge repositories actionable via BRMS engines;**
- **Accelerates and automates the deployment and management of domain business rules;**



Benefits of SIK adoption

- Adds visibility to the way that current software defines and processes business rules;
- Creates the ability to integrate and unify different silos of knowledge;
- Provide knowledge management with the ability to respond in real time and to take action correspondingly;
- Uses the standards.



Conclusion

- **With a SIK adoption, the business rules become machine-accessible and human meaningful**





Mulțumesc!

Thanks You !

Merci!

Dank!



References:

- ◆ Albin, S. (2003) *The Art of Software Architecture: Design Methods and Techniques*, John Wiley & Sons, Books24x7 <http://common.books24x7.com/book/id_6020/book.asp>, pp 1-20
- ◆ Allemang, D. and Hendler, J. (2008) *Semantic Web for the Working Ontologist: Modeling in RDF, RDFS and OWL*, Elsevier, Morgan Kaufmann Publ, pp 1-178
- ◆ Aman-ul-haq and Babar, M. A. (2009) *Tool support for automation architectural knowledge extraction*, University of Limerick Ireland, Shark'09, May 16, 2009 Vancouver, Canada, pp 49-56
- ◆ Antoniou, G. and van Hamerlan, F. (2008) *Semantic Web Primer*, Second Edition, The MIT Press, Cambridge Massachusetts, London, England, pp 1-185
- ◆ Avram V. and Avram D. (2007) *Managing knowledge within Small and Midsized Companies*, Informatica Economică Journal, Vol. XI/No. 4 (44)/2007, ISSN: 1453-1305, pp 33-36
- ◆ Avram, V. (2009) *Should software architecture change to adapt to the knowledge era?*, 4th International Conference Knowledge Management - Projects, Systems and Technologies, Academy of Economic Studies and the National Defense University 'Carol I', ISBN: 978-973-663-783-4, pp 75-77, Bucharest, Romania, pp 75-78
- ◆ Babar, M.A. and Gorton, I. (2007) *Architecture Knowledge Management: Challenges, Approaches, and Tools*, 29th International Conference on Software Engineering (ICSE'07 Companion), 0-7695-2892-9/07 IEEE Computer Society, pp 170-171
- ◆ Berners-Lee, T., Connolly, D. and Swick, R.R. (1999) *Web Architecture: Describing and Exchanging Data*, W3C Note 7 June 1999, www.w3c.org/1990/04/WebData
- ◆ Bass, L., Clements, P. and Kazman, R. (2003) *Software Architecture in Practice*, Second Edition, Addison-Wesley Professional, <http://acmsel.safaribooksonline.com/0321154959>, pp 1-17
- ◆ Cardoso, J. (ed.) (2007) *Semantic Web Services: Theory, Tools, and Applications*, IGI Global, pp 1-21
- ◆ CORTICON Technologies, Inc. (2009) *The Knowledge Pyramid: A Framework for Understanding the Challenges and Opportunities of Managing Decisions as an Enterprise Asset*, www.corticon.com, pp 1-4
- ◆ Daconta, M. C. and Obrst, L., Smith, K. (2003) *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*, John Wiley & Sons, Books24x7 <http://common.books24x7.com/book/id_6073/book.asp>, pp 1-116
- ◆ Davenport, T. and Prusak L. (1998) *Working Knowledge: How Organizations Manage What They Know*, Boston: Harvard Business School Press, pp 1-178
- ◆ Graham, I. (2007) *Business Rules Management and Service Oriented Architecture: a patternlanguage*, John Wiley & Sons, pp 1-235
- ◆ Klyne, G., Carroll, J. (ed.) (2004) *Resource Description Framework (RDF): Concepts and Abstract Syntax*, W3C Recommendation 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>
- ◆ Powers, S. (2003) *Practical RDF*, O'Reilly, (e-Book), pp 1-350
- ◆ Russ, M. (ed.) (2010) *Knowledge Management Strategies for Business Development*, Business Science Reference, Hersey, New York, pp 261-273
- ◆ Segaran, T., Evans, C. and Taylor, J. (2009) *Programming the Semantic Web*, O'Reilly, pp 1-126



=>Data, Information, Knowledge, and Wisdom

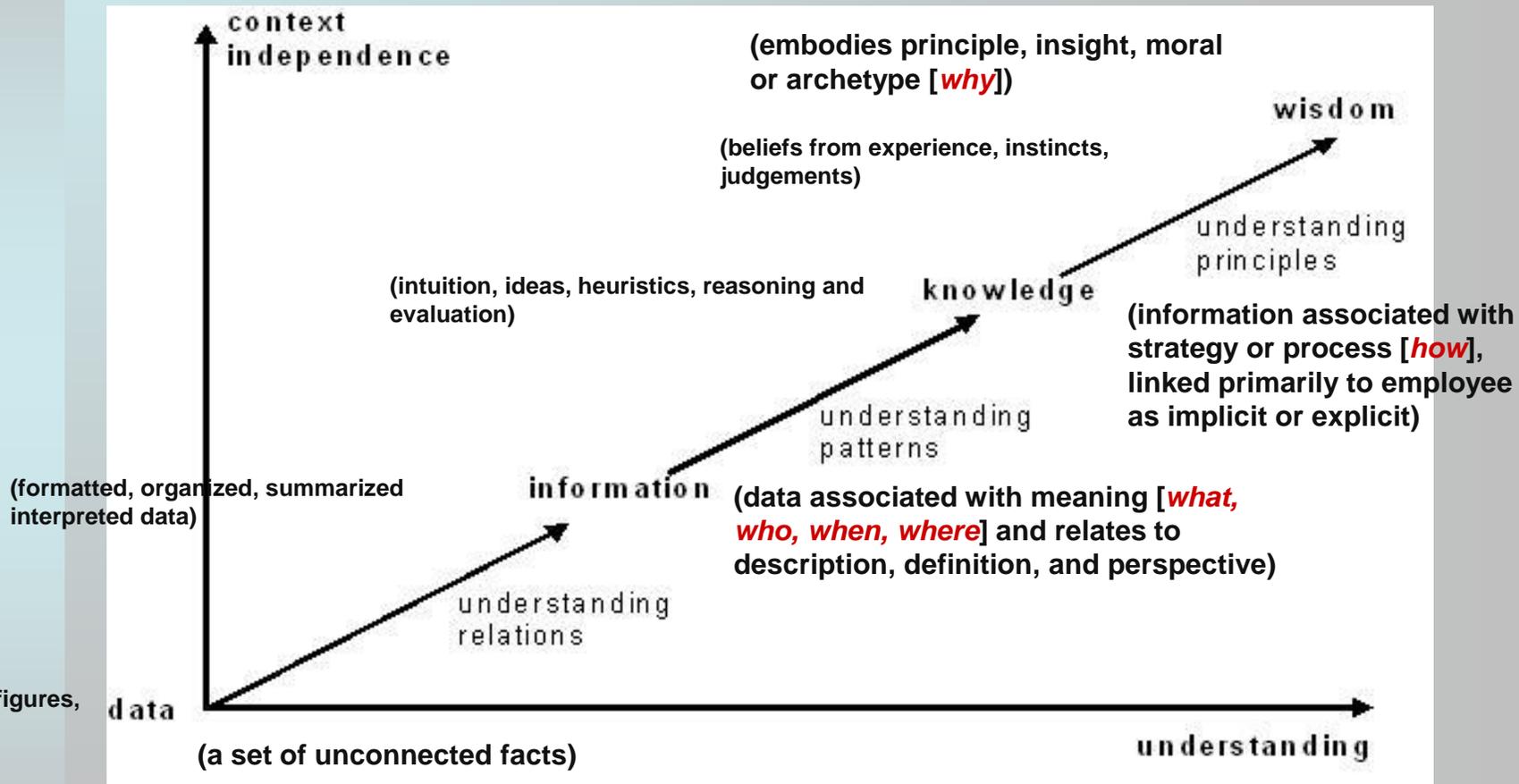


Figure 3: The relationships between data, information, knowledge, and wisdom considering the context independence and understanding

(Source: Neil Fleming, Coping with revolution: Will the Internet change learning? Lincoln University, Canterbury, New Zealand)



Knowledge Management

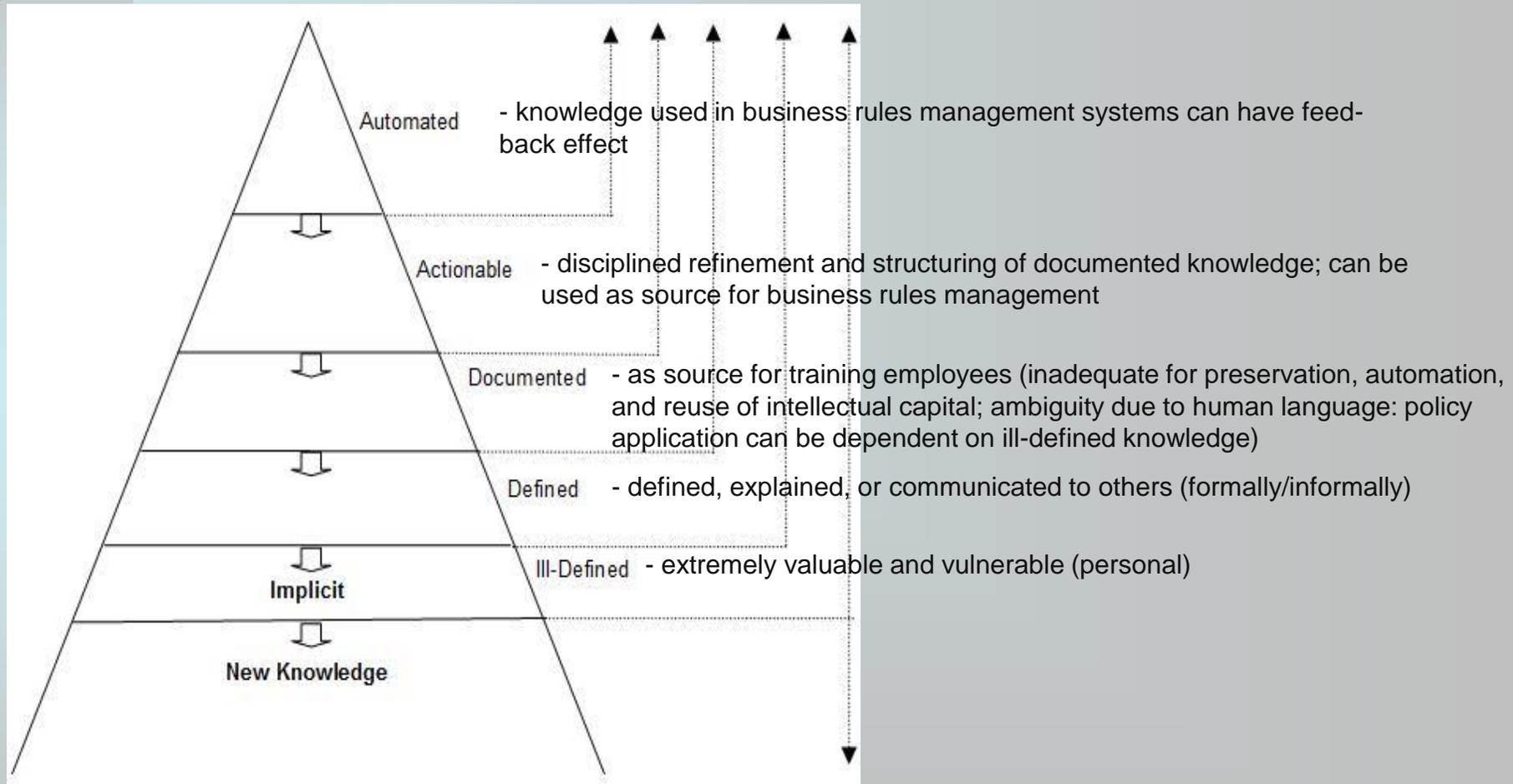


Figure 4: The Knowledge Pyramid

(Source: CORTICON Technologies, Inc., www.corticon.com)