December 2004

Partial Inheritance in Sysperanto, an Ontology of Information Systems

Steven Alter
University of San Francisco

Partial Inheritance in Sysperanto, an Ontology of Information Systems

Steven Alter
University of San Francisco
alter@usfca.edu

ABSTRACT

An offshoot of research on developing methods typical business professionals can use to analyze systems for themselves, Sysperanto is being developed as an ontology codifying concepts and knowledge useful in describing and analyzing systems in organizations. Sysperanto's architecture is organized around the nine elements of the work system framework and the observation that information systems, projects, supply chains, ecommerce, and other important types of systems can be modeled as special cases of work systems. These supertype-subtype relationships provide an opportunity to organize relevant concepts economically based on the "partial inheritance conjecture," whereby most, but not all, elements, properties, and propositions for a specific work system type are inherited by more specialized work system types. This paper explains why Sysperanto or other ontologies with similar purposes require the use of partial inheritance rather than strict hierarchy.

Keywords

Ontology of information systems, partial inheritance, information systems concepts, systems analysis, architecture, work systems, Sysperanto

INTRODUCTION

Sysperanto is being developed as an ontology codifying concepts and knowledge useful in describing and analyzing systems in organizations. Although motivated by research on developing systems analysis methods for business professionals, the codification in Sysperanto may help in understanding the nature of the IS discipline as a conglomeration of partly overlapping but seemingly immiscible slices of different disciplines and techniques. Sysperanto's architecture is organized around the elements of the work system framework and supertype-subtype relationships involving special cases of work systems such as information systems and projects. Sysperanto's architecture attempts to organize concepts economically based on the "partial inheritance conjecture," whereby most, but not all, elements, properties, and propositions for a specific work system type are inherited by more specialized work system types. This paper explains why partial inheritance is essential for Sysperanto or related ontologies, whether or not current ontology managers such as Protégécan handle partial inheritance conveniently. Identifying usable workarounds for handling partial inheritance is one of the challenges of incorporating Sysperanto into a practical vocabulary server or more extensive support tool for systems analysis.

THE WORK SYSTEM FRAMEWORK

The basis of Sysperanto is the concept of work system, a term that has been used by a number of socio-technical researchers and by some practitioners, but apparently in a less specific sense than it will be used here. (For previous uses, see Alter (2003).) A work system (Figure 1) is a system in which human participants and/or machines perform work using information, technology, and other resources to produce products and/or services for internal or external customers. Typical business organizations contain work systems that procure materials from suppliers, produce products, deliver products to customers, find customers, create financial reports, hire employees, coordinate work across departments, and perform many other functions. The work system method is a broadly applicable set of ideas that use the concept of work system as the focal point for understanding, analyzing, and improving systems in organizations, whether or not IT is involved. (Alter, 2002)

The concept of work system is a general case of systems operating within or across organizations. Special cases of work systems include information systems, projects, supply chains, ecommerce web sites, and totally automated work systems. These and other special cases should inherit most of the properties of work systems in general.

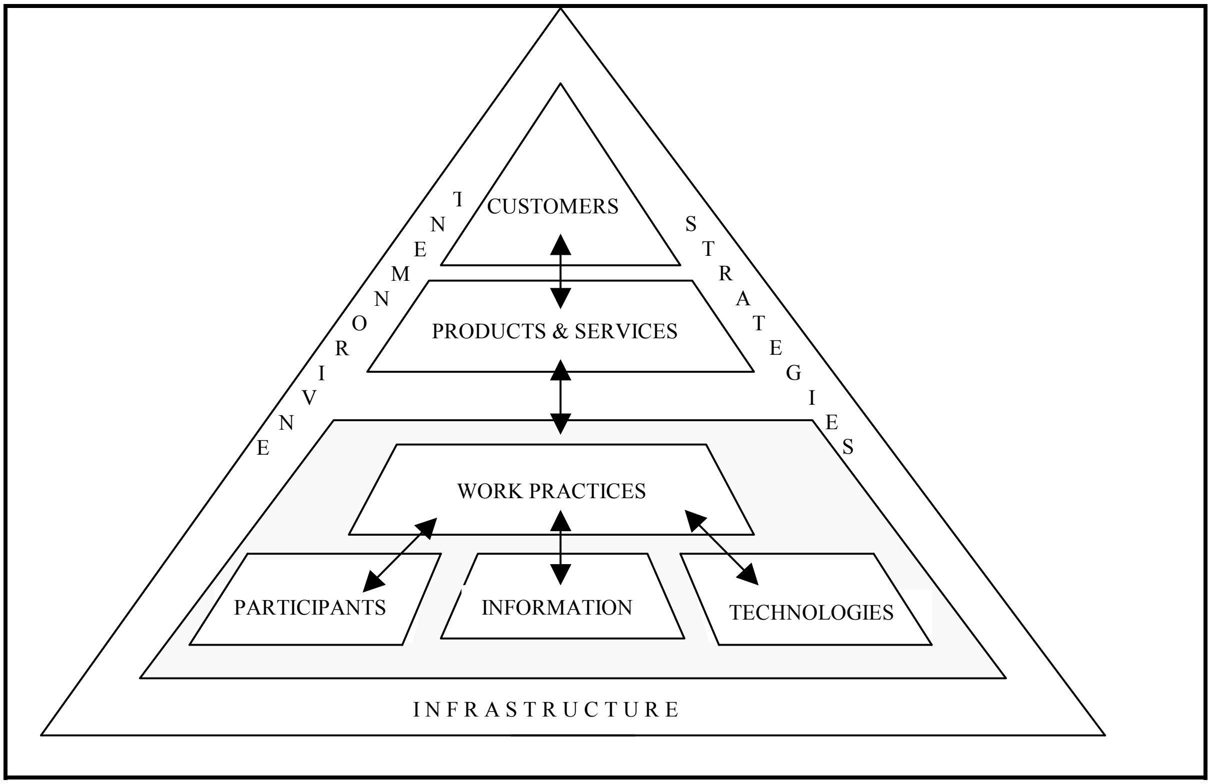

Figure 1. The Work System Framework

SYSPERANTO AS AN ONTOLOGY

Sysperanto is being developed as a theory-based ontology of the IS field. "Ontology is the way we carve up reality in order to understand and process it." (Castel, 2002) "An ontology is an explicit specification of an abstract, simplified view of a world we desire to represent. It specifies both the concepts inherent in this view and their interrelationships. A typical reason for constructing an ontology is to give a common language for sharing and reusing knowledge about phenomena in the world of interest." (Holsapple and Joshi, 2002). Major uses of ontology include communication between humans and/or computer systems, computational inference, and reuse and organization of knowledge. (Gruninger and Lee, 2002). Sysperanto fits in the middle of a dimension that Smith and Welty (2001) propose for comparing different types of ontologies based on complexity and extent of automated reasoning. Sysperanto's underlying theoretical viewpoint imbues a greater degree of organization than a catalog, glossary, or thesaurus, is structured enough to support computerized aids for human analysis processes, but is not directed at automatic inference (e.g., a DSS is a type of system and systems have purposes, therefore a DSS has a purpose).

As the ontology underlying the work system method, Sysperanto's goal is to support understanding and analysis of systems by people, communication between people, and organization and reuse of knowledge about systems in organizations. Therefore any implementation of Sysperanto in software should satisfy the following aspirations: usability by different people at various degrees of depth; relevance to most systems in organizations; inclusion of socio-technical issues; recognition of adaptations, exceptions, and workarounds; and acceptance of slightly vague or partially overlapping concepts

THE PARTIAL INHERITANCE CONJECTURE

The architecture of Sysperanto is organized around the elements of a work system and the fact that information systems, projects, supply chains, and other important types of systems can be modeled as work systems. These supertype-subtype relationships provide an opportunity to organize relevant concepts economically through the use of partial inheritance.

According to the "partial inheritance conjecture" most (perhaps 80% or more) of the elements, properties, and propositions for a specific work system type are inherited by more specialized work system types. In other words, most, but not all, of the ideas and vocabulary relevant to special cases of information systems such as MIS or CRM or "knowledge management systems" are inherited from information systems in general and, further, from work systems in general. Although different names may be used for certain elements or properties in the context of a more specialized work system type, partial inheritance relationships is useful in organizing knowledge within and across work system types because most of the knowledge can be associated with the more general work system type and inherited by its subtypes. As a specific example, Figure 2 represents the conjecture that most success factors of work systems are inherited by subtypes (information systems and projects), and that most success factors for those subtypes are inherited by the next level of subtypes (specific types of IS and specific types of projects). Table 1 illustrates the underlying reality by listing a number of common success factors often related to the success of information systems, projects, or special types of either. Every one of these success factors applies to work systems in general and can be viewed as part of the inheritance from work systems in general to its special cases.

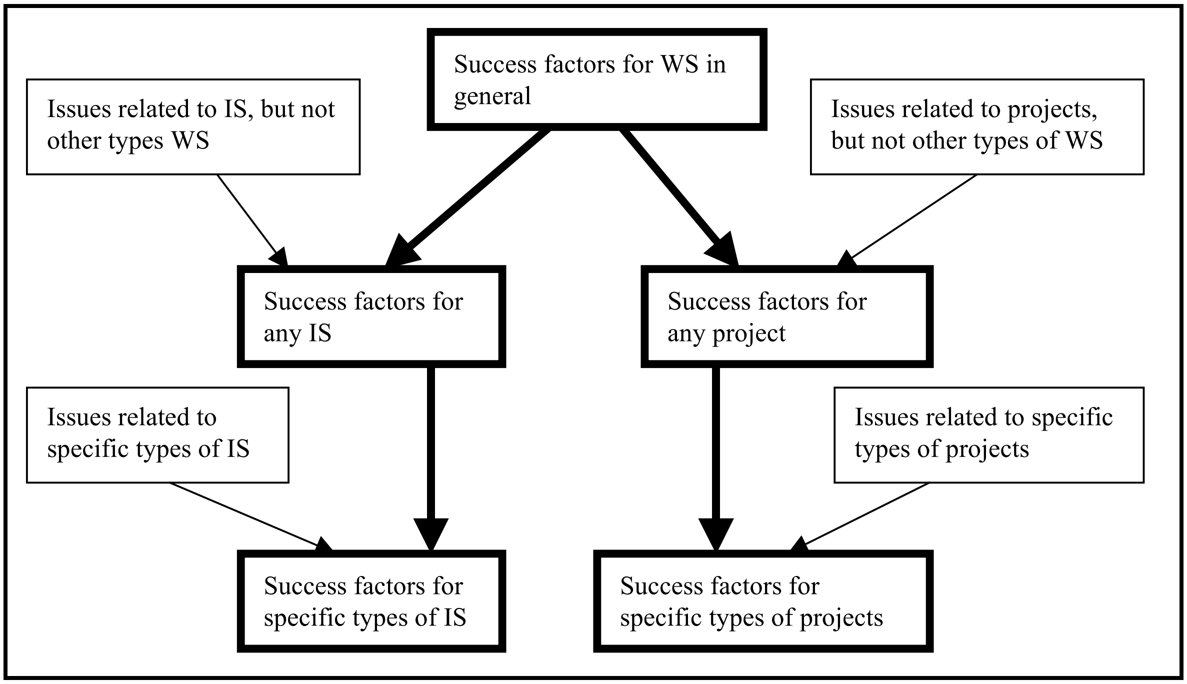

Figure 2: Inheritance of Success Factors by Special Cases of Work Systems

Work system element	**Work system success factors that are also pertinent to special cases such as information systems and projects**
Work practices	Fit of business process with other elements Adequate resources for work practices Effective operational management
Participants	Appropriate skills and understanding Interest in doing this type of work Motivation to do this work in this setting Ability to work together to resolve conflicts
Information	Adequate information quality Adequate information accessibility Adequate information presentation Adequate information security
Technology	Ease of use (for IT or other technologies) Adequate technology performance (“horsepower”) Maintainability and compatibility
Customers and products/services	Product design consistent with customer needs Adequate product performance

Infrastructure	Adequate technical infrastructure Adequate human infrastructure
Environment	Management support Consistency with culture Cooperative decisions about work methods Low level of turmoil and distraction
Strategies	Fit between work system strategy and organization strategy

Table 1: Examples of Success Factors Inherited from Work Systems in General by Special Cases of Work Systems such as Information Systems and Projects

An additional conjecture may explain why it is so difficult to generalize about information systems. The "level-skipping conjecture" says that the various types of information systems (MIS, TPS, DSS, CAD, ERP, CRM, and so on) differ so greatly in form and function that information systems in general have few properties in common beyond those inherited from work systems in general. Accordingly, most of the properties of information systems in general come from work systems in general, few additional properties are related to information systems in general, and a larger number of properties are related to the unique features of the special cases. If this conjecture is largely correct, the frequently discussed quest for a unique body of knowledge for the IS discipline will not succeed.

META-MODEL

Many of the frustrations with the current IS discipline reflect its existence as a loose, unsettled conglomeration of partly overlapping but seemingly immiscible slices of different disciplines and techniques including conceptual modeling, organization behavior, total quality management, human communication, coordination theory, information theory, computer science, and microeconomics, among many others.

Sysperanto is designed around a meta-model (Figure 3) motivated by the conjectures above and by the recognition that people understand business and organizational reality by slicing it in multiple ways. The "work system" is the meta-model's basic unit for understanding any particular system in an organization. Concepts and generalizations about work systems, are organized around work system types, starting with the most general type, "work systems in general." Figure 3 says that any *work system type* might have multiple subtypes. Instances of any work system type are summarized using work system *elements* that are relevant to that type. For example, the elements for work systems in general are the nine elements in the work system framework (Figure 1). Each element for any work system type is understood through a series of *slices*. The vocabulary and knowledge for each slice consists of *properties* applicable to any instance of that type. The properties themselves may be components or phenomena (nouns), actions or functions (verbs), characteristics (adjectives), performance variables (adverbs), relationships, phenomena, and generalizations. Each part of Figure 3 will be explained in more depth.

WORK SYSTEM TYPES

As is implied by the different levels in Figure 3, distinctions between work system types are based on:

- inclusion or exclusion of specific elements
- relevance or irrelevance of specific slices
- relevance or irrelevance of specific properties within slices
- specification of particular values or ranges of values for particular properties.

For example, a project is a work system that will cease to exist after it produces certain products. Similarly, an information system is a work system whose work practices involve only processing of information and do not include communication, social relationships, thinking, or physical activity not directly related to processing of information. Supply chains are interorganizational work systems whose participants include suppliers and their customers and whose work practices are devoted to establishing and fulfilling customer requirements.

The nine elements in the work system framework (Figure 1) are included in even a rudimentary understanding of the most general case, "work systems in general." The same nine elements apply to closely related subtypes such as information system, project, and supply chain. More distant subtypes of work systems might add or eliminate elements. "Work system as a whole" is treated as a tenth element in the meta-model because some properties of work systems (such as age, geographic dispersion, and degree of competitive significance) apply to the entire work system rather than to just to specific elements such as the work practices or technology.

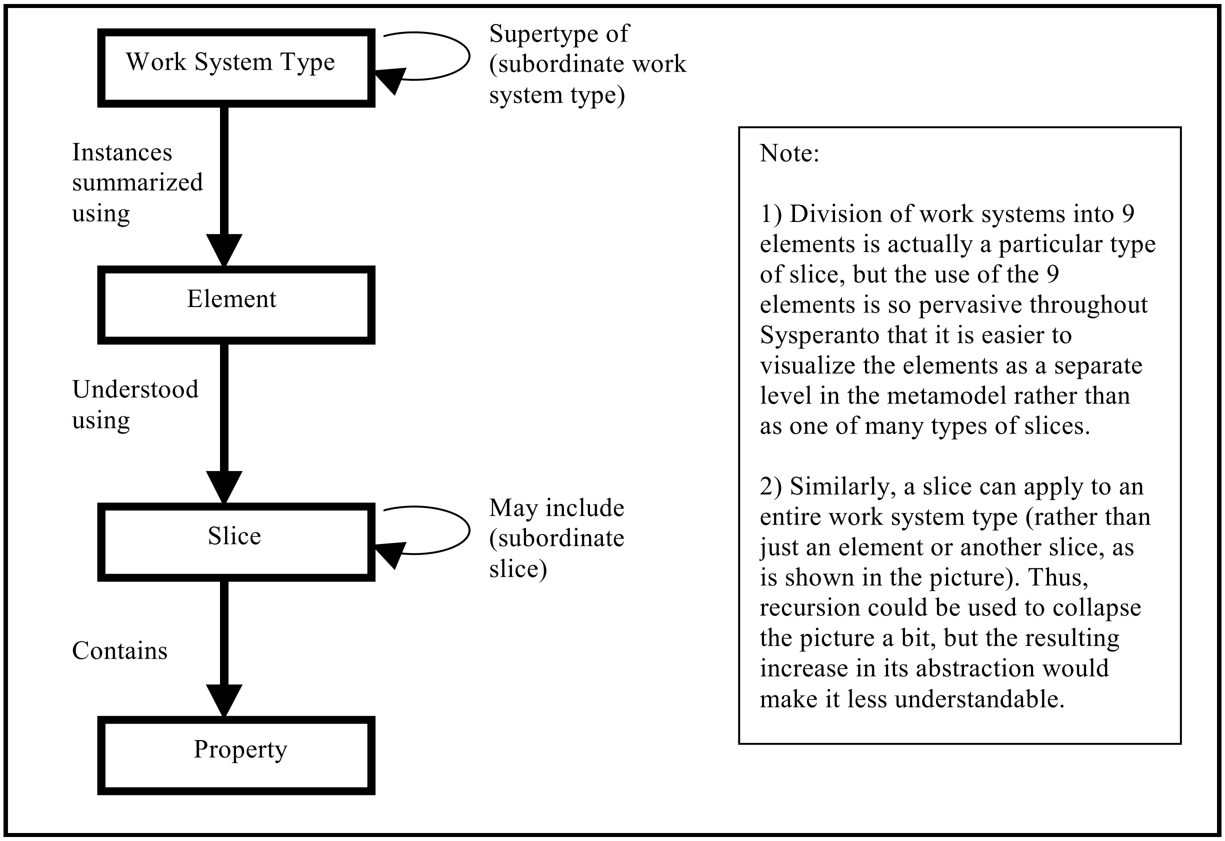

Figure 3: Structure of Sysperanto's Meta-Model

SLICES

Disciplines, subdisciplines, and areas of knowledge can be viewed as particular ways to slice reality in order to understand it. For example, a particular business situation can often be viewed as a decision situation or as a communication situation, with each viewpoint bringing valid and useful concepts that may help in understanding the situation. In some cases, the concepts overlap, such as when decision making involves communication and communication involves decision making.

In Sysperanto a slice is a related set of properties that can be applied when trying to understand or analyze a particular work system. For convenience, all of the properties of a work system element (or of the entire work system) are grouped within specific slices. In most real world situations, many different slices are relevant, and some of those slices may overlap by invoking similar or related concepts.

Table 2 lists the slices in the current version of Sysperanto. As Sysperanto develops further, additional slices will be added for the elements themselves, for "sub-slices" within a slice for a particular work system element, and for the elements of special cases such as information systems, projects, and supply chains.

Element of a Work System	**Slices Relevant to this Element**	**Comments about the Slices for this Element**
Work practices	• Work practices as a whole • Business process • Communication	• *Work practices as a whole*: properties concerning work practices as a whole, rather than more specialized slices. • *Business process, communication, sense making, decision making, coordination, information processing, thinking, physical actions*: Each of these slices brings concepts and generalizations that apply to the

	• Sense making • Decision making • Coordination • Information processing • Thinking • Physical actions	work practices in many situations, but not all.
Participants	• Groups or individuals • Roles • Impacts on participants • Impacts of participants	• *Groups or individuals*: properties concerning participants viewed as groups of people or individuals, without reference to work practices • *Roles*: properties related to participant roles, without specific reference to identity or characteristics of individuals. • *Impacts on/ of participants*: properties concerning participants that are related to impacts on or by work practices
Information	• Information as a whole • Database • Document • Conversation • Knowledge • Workspace signals and cues	• *Information as a whole*: properties related to all or a subset of the information in a work system, without separately referring to characteristics of different types of information such as database or document. • *Database, document, conversation, knowledge, workplace signals and cues*: Even though the growth of multimedia has generated some disagreement about basic definitions of terms such as database or document, some separate concepts exist for each of these types of information. Also, although conversation might not seem to belong in the same category as database, documents, or knowledge, much of the information in meetings and other coordination activities can be viewed as conversation consisting of speech acts.
Technologies	• Technology as a whole • Artifacts (tools) • Techniques • Interfaces	• *Technology as a whole*: properties related to all or a subset of the technologies in a work system, without separately referring to properties of particular types of technologies. • *Artifacts* and *techniques*: Technologies can be viewed as artifacts, techniques, or techniques inscribed on artifacts. Most of the concepts for technologies may be related to artifacts or techniques rather than technologies as a whole. • *Interfaces*: Technologies typically have interfaces through which users guide or receive information from the technology.
Products and services	• Products and services as a whole • Physical product • Information product • Service • Social product • Intangible product • Customers experience	• *Products and services as a whole*: properties related to all or a subset of the products and services produced by a work system, without separately referring to properties of particular types of products. • *Physical product, information product, service, social product, intangible product, customer experience*: Some important concepts refer specifically to each of these aspects of the products and services produced by a work system.
Customers	• Groups or individuals • Roles • Impacts on customers as individuals or groups • Impacts of customers as	• *Groups or individuals*: properties concerning customers as groups of people or individuals • *Roles*: properties related to customer roles, without specific reference to identity or characteristics of individuals. • *Impacts on/ of customers*: properties concerning customers that are

	individuals or groups	related to impacts on or by products and serviced the system produces
Infrastructure	• Infrastructure as a whole • Human infrastructure • Information infrastructure • Technical infrastructure	• *Infrastructure as a whole*: properties related to all or a subset of the infrastructure supporting a work system, without separately referring to properties of particular types of infrastructure. • *Human, information, and technical infrastructure*: Some separate concepts may apply for each of these.
Environment	• Environment as a whole • Culture • Political environment • Competitive environment • Policies and procedures • Regulations/ standards • History	• *Environment as a whole:* properties related to all or a subset of the environment surrounding a work system, without separately referring to properties of particular aspects of that environment. • *Culture, political environment, competitive environment, policies and procedures, regulations and standards, history:* Separate concepts apply for each of these separate aspects of the environment.
Strategies	• Strategies as a whole • Work system strategy • Organization's strategy • Firm's strategy	• *Strategies as a whole*: properties related to all or a subset of the strategies that affect a work system, without separately referring to properties of particular levels of those strategies. • *Strategy of the work system, organization, and firm*: Some separate concepts may apply for each of these separate levels of the relevant strategies.
Work system as a whole	• Work system as a whole	• *Work system as a whole*: Although most slices and properties are related to elements of work systems, some are related to a work system as a whole. For example, a work system's age, criticality, fragility, and capacity are actually related to the work system as a whole rather than its individual elements.

Table 2: Slices Included in the Current Version of Sysperanto

PROPERTIES

A property of a work system may be a mathematically derived quantity (e.g., average age or diversity of participants), a qualitative judgment about the inherent nature of an aspect of the system (e.g., degree of structure or complexity), or an emergent property that depends on interactions (e.g., group cohesiveness). For convenience, each property of a work system element (or of the entire work system) is assigned to a particular slice.

To illustrate the different types of properties, Table 3 identifies different types of properties and gives examples within each type for the *work practices as a whole* slice of the *work practices* element of *work systems in general*. Among the various types of properties, components and phenomena are like nouns; actions or functions are like verbs; characteristics are like adjectives (inherent features related to form and structure that tend to persist until changed); performance variables are like adverbs (describing how well something was done or what its status was during a particular event or time interval); and relationships involve topics linking two or more things (such as overlaps, dependencies, complementarities, compatibility, and interoperability). For work system subtypes such as information system or project, the *work practices as a whole* slice of the *work practices* element will inherit many of the properties in Table 3, but might also have other properties that are not properties of work systems in general.

Table 3 presents properties related to the *work practices as a whole* slice for the *work practices* element of a work system. Other slices for work practices, such as business process, communication, sense making, coordination, and decision-making (see Table 2) would add other properties for each slice. For example, the communication slice would add *degree of social presence* and *degree of synchronicity* as two characteristics of a communication situation and would add *clarity* and *degree of absorption by the recipient* as performance variables. These properties apply in most communication situations even though they would not be considered characteristics or performance variables for work systems in general.

Property Type	**Examples of properties for the *work practices as a whole* slice of the *work practices* element of *work systems in general***
Components and phenomena	• Norms and values inherent in the work practices • Mutual awareness between participants • Pre-computation (creating checklists, plans, etc., so that work system participants won't need to reinvent the wheel continually)
Actions or functions	• Initiate (trigger) • Plan • Set-up • Execute • Repair/ rework • Complete (packaging for next step or for customer) • Control (feedback) • Coordinate • Track (perform record keeping) • Manage
Characteristics	• Degree of structure • Range of involvement • Level of integration • Complexity • Rhythm • Degree of reliance on machines • Attention to planning and control • Formality of exception handling • Error-proneness • Fault tolerance • Degree of variety, repetitiveness • Degree of improvisation • Degree of interruption
Performance variables	• Rate of activity • Rate of output • Consistency • Productivity • Speed • Down time • Security • Rate of rework • Value added
Relationships	• Produces for: Customer, Upstream process • Receives from: Supplier, Downstream process

	• Initiated by: Triggering event • Governed by: Rule, policy • Limited by: Constraint • Dependent on: another process or something else • Interdependent with: another process or something else
Change-related actions	• Adaptation • Workaround • Experiment
Change-related characteristics	• Adaptability (vs. rigidity) • Stability • Scalability
Change-related performance variables	• Amount of effort to make a change • Amount of effort in relation to the extensiveness of a change
Generalizations	• If the degree of structure is too high, work system participants are prevented from using their judgment; if it is too low, foreseeable errors are more likely because rules are applied inconsistently. • If complexity is too high, participants may not understand what they are doing or why; if it is too low, work practices may not distinguish between cases that should be handled differently.

Table 3: Property Types and Examples of Related Properties for the *Work Practices as a Whole* Slice of the *Work Practices* Element

PARTIAL INHERITANCE

Figure 2 illustrated the idea of inheritance by showing that success factors for work systems in general should be inherited by special cases of work systems such as information systems and projects. Each special case might have additional success factors related to their unique properties. For example, the success factor *clear milestones* applies to projects in general but does not apply to work systems in general.

Figure 4 extends the meta-model characterization in Figure 3 by illustrating the role of partial inheritance, according to which subtypes inherit some, but not necessarily all, elements, slices, and properties from supertypes. The relative size of the boxes for inherited and non-inherited elements, slices, and properties in Figure 4 reflects a belief that most of the important slices and properties for subtypes are inherited from the most similar supertypes and that relatively few slices or properties relevant to supertypes are not inherited by subtypes. Additional slices and properties may also be relevant due to the unique nature of each special case. For example, the process of defining *international MIS* as a work system type in Sysperanto would start by assuming that the slices and related properties for the work system type *MIS* would apply, and then, based on whatever is believed unique about international MIS, identifying whatever slices and properties for MIS in general are inapplicable and what additional slices or properties need to be added for international MIS.

CONCLUSION: THE NEED FOR PARTIAL INHERITANCE

Hierarchy, partial inheritance, and multiple inheritance have been discussed (e.g., Nierstrasz, 1988; Zhu and Zhou, 2003; Krogh et al, 1996) and used extensively in relation to theories of knowledge and object oriented programming theory and languages. In relation to Sysperanto, partial inheritance raises two main questions. The first question is whether partial inheritance is necessary for achieving Sysperanto's twin goals of supporting systems analysis and codifying IS concepts and knowledge. The second question is whether partial inheritance can be implemented conveniently using existing ontology managers such as Protégé, which are based on *is-a* relations.

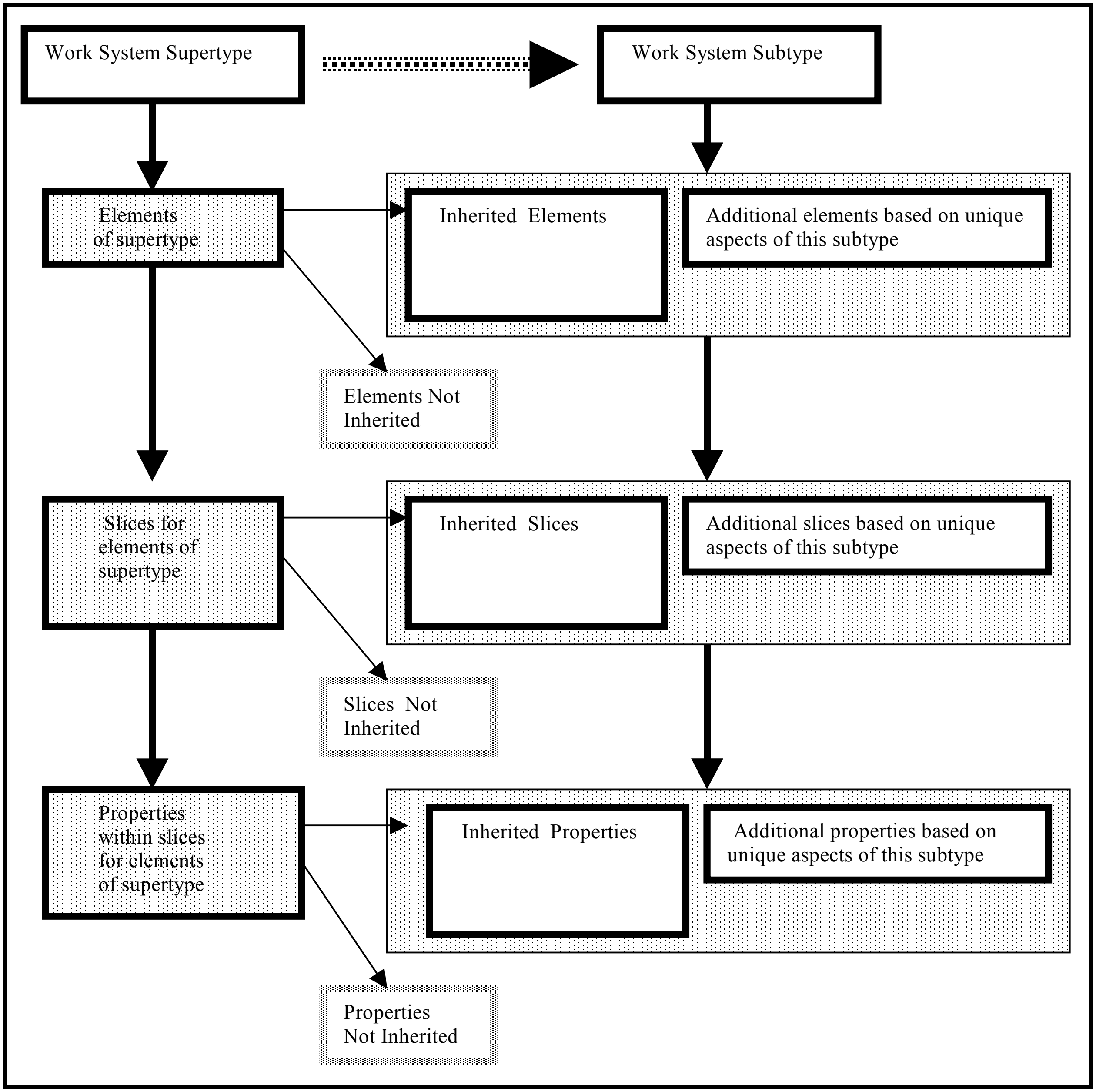

Figure 4: Partial Inheritance

The necessity of using partial inheritance is a straightforward consequence of the problems Sysperanto addresses. For example, Table 2 listed slices for each of nine elements of the work system framework. Each slice encompasses concepts that are relevant to work systems in general, and some of the properties within one slice overlap with those of another. For example, *communication* and *decision making* are slices of work practices that contain a rich array of concepts, but that also overlap because decision making involves communication and vice versa. It might be possible to sort the properties in a way that eliminates overlap between the slices, but the nature of the topics in Table 2 implies that attempts to fit the hierarchical structure of ontology managers would probably generate artificial hierarchies that are counter-intuitive and cause confusion. From a usability viewpoint it is much simpler and clearer to accept overlaps between slices instead of trying to create artificial hierarchies.

If Sysperanto is to have a genuine impact, computerized storage of properties and context-related access to relevant properties within Sysperanto should be both efficient and effective. Efficiency requires eliminating redundant storage of individual properties and data about those properties, such as their definitions, synonyms, references to the literature, and examples of usage. For example, the concept *speed* applies to work systems in general, information systems, and projects. All data about the property *speed* should be stored at the work system level and inherited by the more specialized levels. In contrast, all data about the property *requirements creep* should be stored at the level of projects in general because requirements creep is broadly relevant to projects but not broadly applicable to work systems or information systems in operation. Maintaining data about each property will be challenging enough without redundancy. With redundancy it would become a nightmare.

Attainment of this type of efficiency depends partly on conceptual architecture and partly on the tools that are used. During initial development, Sysperanto's structure and vocabulary have been stored in Microsoft Word tables. Although this approach has been reasonably efficient for developing the initial version of the top-level vocabulary, neither Word nor Excel nor Access provides an effective method for handling both recursive inheritance relationships and partial inheritance across type-hierarchies. Ontology managers are designed to handle inheritance relationships, but partial inheritance conflicts with the strict hierarchy in the *is-a* relationships at their core. The technical approach in using these tools may turn out to be a workaround that handles inheritance and type-hierarchies awkwardly. For example, in Protégé it is possible to identify particular inherited properties as "hidden." The ramifications of using this type of workaround will not be known until a multi-layered version of Sysperanto is loaded into Protégé or other similar tools. Regardless of which tools are used initially, the efficiency of Sysperanto or other ontologies with similar purposes will depend on handling partial inheritance in a convenient and efficient manner.

On the usage side, effectiveness for analysis and/or design implies relative ease in finding and using the relevant properties during an analysis and/or design process. Research to date on the work system method has demonstrated that the use of systems analysis templates at the work system level can support a basic level of effectiveness. It is possible to produce templates for different types of information systems or projects, but just producing and maintaining these templates could involve a lot of effort and would run counter to using the structure of the concepts and knowledge directly.

Maximizing effectiveness calls for interactive, context-sensitive tools that help in accessing whatever slices and properties are truly relevant to a particular stage in the analysis of a particular situation. Decision support tools of this type could help business and IT professionals define and explore system-related problems, opportunities, and potential directions for improvement. Following those discussions IT professionals might use UML and other tools designed to document and verify the details of computer applications.

Regardless of what form Sysperanto-based tools take, partial inheritance will play an integral role in both internal storage of the slices and properties and presentation of relevant slices and properties to users. Inadequate treatment of partial inheritance would result in internally inefficient tools more likely to present irrelevant concepts or overlooking relevant concepts.

REFERENCES

1. Alter, S. (2002) "The Work System Method for Understanding Information Systems and Information Systems Research," *Communications of the AIS* 9, 9, 90-104.
2. Alter, S. (2003) "18 Reasons Why IT-Reliant Work Systems Should Replace 'The IT Artifact' as the Core Subject Matter of the IS Field," *Communications of the AIS,* 12, 23, 365-394.
3. Castel, F. (2002) "Ontological Computing," *Communications of the ACM*, 2002, 45, 2, 29-30.
4. Gruninger, M. and Lee, J. (2002) "Ontology: Applications and Design," *Communications of the ACM*, 45, 2, 39-41.
5. Krogh, B., Levy, S., Dutoit, A., and Subrahmanian, E. (1996) "Strictly Class-Based Modeling Considered Harmful," *Proceedings of the Hawaii International Conference on System Sciences (HICSS'29).* Maui, Hawaii, viewed on Feb. 3, 2004 at http://www.ndim.edrc.cmu.edu/papers/strictly.pdf
6. Holsapple, C. W. and Joshi, K. D. (2002) "A Collaborative Approach to Ontology Design," *Communications of the ACM*, 45, 2, 42-47.
7. Nierstrasz, O. (1989) "A Survey of Object Oriented Methods," pp. 3-21 in W. Kim and F. Lochovsky, eds., *Object-Oriented Concepts, Databases, and Applications*, ACM Press and Addison Wesley.
8. Smith, B. and Welty, C. (2001) "Ontology: Toward a New Synthesis," *Proceedings of FOIS '01*, Oct. 17-19, 2001, Ogunquit, ME.
9. Zhu, H. and Zhou, M. (2003) "Methodology First and Language Second: A Way to Teach Object-Oriented Programming," OOPSLA '03, Oct. 26-30, Anaheim, CA.