H. Delugach and B. Lampkin, "Troika: Using Grids, Lattices and Graphs in Knowledge Acquisition," in *Working with Conceptual Structures: Contributions to ICCS 2000*, G. Stumme, Ed. Aachen, Germany: Shaker Verlag, 2000, pp. 201-214.

Troika: Using Grids, Lattices and Graphs in Knowledge Acquisition

Harry S. Delugach and Brian E. Lampkin
Computer Science Department
Technology Hall N-300
The University of Alabama in Huntsville
Huntsville, AL 35899
(256) 890-6614
Email: delugach@cs.uah.edu

Abstract

A knowledge acquisition technique called Troika is presented. Based on a combination of repertory grids, conceptual graphs and formal concept analysis, Troika is a hybrid approach that combines these three approaches. The approaches are introduced and their strengths summarized. Troika's basic algorithms are presented. Some samples from an actual acquisition process are presented to provide some flavor of the approach.

1 Introduction

Knowledge engineers want to improve their methods of acquiring, transferring, analyzing and representing knowledge. In attempting to automate solutions to knowledge-intensive problems, the knowledge engineer (KE) must endeavor to capture information and experience from people who are experienced in particular domains of interest. Such knowledge acquisition (KA) techniques are numerous, usually based on ad-hoc informal question-and-answer interactions with experts. In general, current KA techniques tend to be either too broad and weak or else too narrow and specific for effective elicitation and representation in general knowledge bases.

This paper presents a hybrid technique called Troika that overcomes some of these weaknesses. Our work is motivated by the observation that existing knowledge acquisition techniques each possess weaknesses that hinder their effectiveness in eliciting all the required knowledge for any possible domain. This hybrid will be comprised of essential parts of three approaches – repertory grids, conceptual graphs, and formal concept lattices – so that they complement each other's effectiveness.

2 Background

In this section, we will briefly summarize the KA process and the three approaches on which we have based our work.

Knowledge acquisition (KA) is the process of transferring and transforming information from some knowledge source to a computer. The study of KA is broad and varied and many books discuss the details of this area, e.g. [1, 2]; a complete summary is clearly beyond the scope of this paper. We note only the key terminology here.

Much of the KE's work is manual and slow. For this reason, some researchers describe the process as the "KA bottleneck" in the development of knowledge-based systems [1]. Interviewing experts and watching them resolve problems is both timeconsuming and error-prone. In such approaches the KE plays an integral role in the transfer and the transformation of information from the domain experts to the expert system. Recognizing the bottleneck, KE's encourage the development of automated KA techniques to speed the KA process and to ensure its accuracy. Ideally, such tools should work with minimal interference from a KE. These tools should accept information directly from experts and then automatically structure that knowledge appropriately in order to operate as a decision-support tool.

Three goals of KA are acquisition, analysis and representation. At the risk of over-simplifying, we use techniques that are strong in these areas: repertory grids for acquisition, formal concepts for analysis, and conceptual graphs for representation. We assume that the reader is familiar with the basics of conceptual graphs, and summarize the other two approaches with which some readers may be less familiar.

2.1 Conceptual Graphs

Conceptual graphs (CGs), introduced by John Sowa, represent knowledge by concepts and the relations between them. They have been studied extensively through a number of international conferences, most recently [3]. The reader is assumed to be familiar with the basic theory of conceptual graphs. Conceptual graphs have been used extensively for representation and inference, which is to be expected given their basis in first-order logic systems. There are limitations to conceptusl graphs, however, that the other techniques serve well in overcoming.

One limitation of conceptual graphs is their lack of direct support for automated knowledge acquisition – except for relying on an experienced CG analyst to transcribe graphs based on interviews and documents, there is no well-established technique for building conceptual graphs to represent an expert's knowledge. One notable exception is the strategy of automatically populated a CG knowledge base from a database; however, the original template graphs for such an approach are still acquired manually from an experienced analyst.

Another limitation of conceptual graphs is their lack of support for probabilistic reasoning – this is to be expected since they rely on a deontic logic of boolean-valued propositions. Since repertory grids permit acquiring constructs along a spectrum between two poles, they support these "in-between" characteristics. And since formal concept analysis supports self-organizing of concepts among these characteristics (conceptual scaling) they also provide a bridge from the concepts themselves to probabilistic reasoning among those concepts.

2.2 Repertory grids

Repertory grids are a well-known knowledge acquisition and representation technique that was introduced with Kelly's research on personal construct psychology, or PCP [4], [5], [6]. Personal Construct Psychology theorizes that people characterize the universe by associating constructs, or attributes, to each object, or thing, within the universe. In accordance with the theory, the repertory grid technique distinguishes the objects of a problem domain (called *elements*) through their attributes (called *constructs*). A single repertory grid is represented as a matrix whose columns have element labels and whose rows have construct labels. In a sense, a grid is a representation for a class of objects, or individuals. For example, the grid shown in Figure 1 represents an object class called **Employee**, where "+" means the individual has that attribute, "-" means it does not.

Sam	Ira	Sue	Bob	Louise	Juan	Preetha	Marjorie	
-	-	+	-	-	+	+	-	*works on Project A*
-	+	-	+	+	-	-	-	*works on Project B*
+	-	-	-	-	-	-	-	*does clerical work*
+	+	+	+	+	+	+	-	*reports to others employees*
-	-	-	+	-	-	+	+	*employees report to him/her*

Figure 1. Repertory grid representing class "Employee".

The values assigned to an element-construct pair need not be Boolean. Grid values may have numeric ratings, probabilities, etc., where each value reflects the degree to which a construct applies to an element. The possibility to use different kinds of values enriches the repertory grid technique in eliciting and representing knowledge about a domain.

The structure of repertory grids provides a framework for organized methods of eliciting information. For example, one popular process, called triadic elicitation, begins with the selection of three candidate elements such that two of the three elements are chosen, and then similarities and differences are sought between them to elicit attributes.

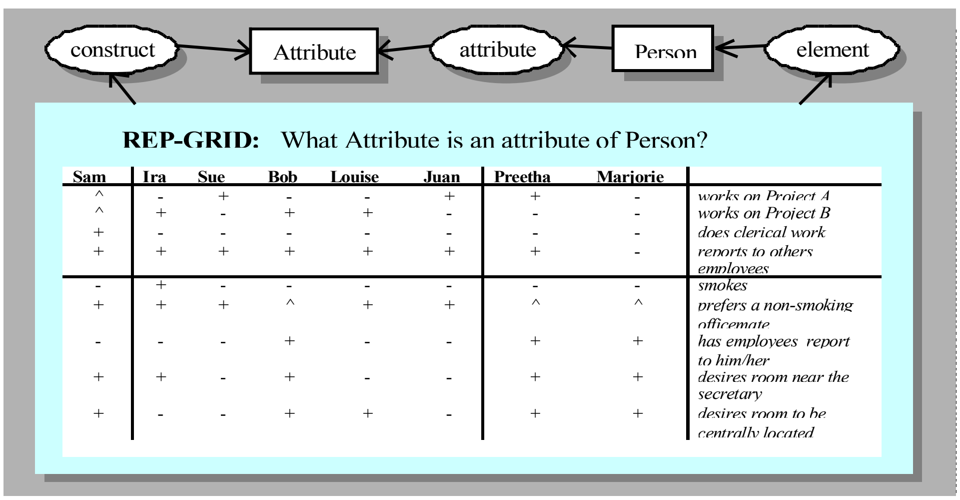
Repertory grids are well-suited for classification among concepts, given even a partial set of characteristics and their respective values. One weakness of repertory grids is their inability to acquire process knowledge or other relationships among concepts; this was a major motivation behind the development of tracked repertory grids [7]. There are now two ways to analyze relationships among multiple grids. One way is *laddering*, which allows a grid's concept to stand in a sub-type or super-type relationship to another grid's concept [5]. . Another way is *tracked repertory grids* [7] that supports showing more generalized associations between repertory grids.

A single repertory grid is a rich representation of a concept. This notion ties repertory girds to conceptual graphs. The elements of a grid constitute the extent of its concept. The set of constructs that apply to these elements forms the intent of the concept. Both of these notions tie formal concept analysis (FCA) to conceptual graphs and repertory grids. As grids can be related to one another, so can their associated concepts be related to one another through a conceptual graph.

A key notion in our work is to embed a repertory grid as a concept in a conceptual graph using a concept type labeled REP-GRID as introduced in [8]. The repertory grid is represented as the literal referent (or, individual) of the REP-GRID type. As an illustration, consider the grid-graph in Figure 2 showing Person related to Attribute. The elements (i.e., the column labels) of the grid are the extension of the Person concept, whereas the constructs of the grid (i.e., the row labels) are the extension of the Attribute concept. The relationship between the graph and the grid can be paraphrased as “the construct of REP-GRID is Attribute” and “the element of REP-GRID is Person”.

2.3 Formal Concept Analysis

As the name implies, the notion of a concept is central to FCA. In a formal context, a set of entities forms the extension of the concept, and the set of attributes is the intension of the concept. FCA characterizes a context as a fixed set of entities and attributes, presented in a cross-reference table. The information found in such a cross-reference table is often depicted as a line diagram, or context lattice. For our work, repertory grids provide the cross-reference table from which one can derive line diagrams and then identify sub-types among the elements of the grid. This is similar to the approach used in [9].



Sam	Ira	Sue	Bob	Louise	Juan	Preetha	Marjorie	
^	-	+	-	-	+	+	-	*works on Project A*
^	+	-	+	+	-	-	-	*works on Project B*
+	-	-	-	-	-	-	-	*does clerical work*
+	+	+	+	+	+	+	-	*reports to others employees*
-	+	-	-	-	-	-	-	*smokes*
+	+	+	^	+	+	^	^	*prefers a non-smoking officemate*
-	-	-	+	-	-	+	+	*has employees report to him/her*
+	+	-	+	-	-	+	+	*desires room near the secretary*
+	-	-	+	+	-	+	+	*desires room to be centrally located*

Figure 2. An example of a grid-graph.

As a brief illustration, consider the repertory grid in Figure 2 that characterizes musical instruments according to some attributes. The set of entities, or instruments, is defined by E={**Flute**, **Cornet**, **Tuba**, **Piano**, **Alto Saxophone**, **Banjo**} and the set of attributes is defined by A={*has mouthpiece*, *has a conical bell*, *has strings*, *has valves*, *play soprano parts*}. The repertory grid represents a concept called Musical Instruments.

Flute	Cornet	Tuba	Piano	Alto Saxophone	Banjo	
X	X	X		X		*has mouthpiece*
	X	X		X		*has a conical bell*
			X		X	*has strings*
	X	X				*has valves*
X	X		X	X	X	*plays soprano parts*

Figure 3. A repertory grid for Musical Instruments.

FCA provides for the cross-reference table to be displayed graphically. The diagram in Figure 4 represents a concept lattice and shows the same information as the cross-reference table in Figure 3. Each node of the lattice represents a subset of elements and of attributes defining the context. Hence, each node is labeled with Element names or Attribute names or both. Observe that element names dominate the nodes in the lower part of the graph while attribute labels dominate the upper nodes in the graph. The top-most node, or *supremum*, represents the set of all elements. The bottom-most node, or *infimum*, represents the null set. Neither the supremum nor the infimum ever have labels.

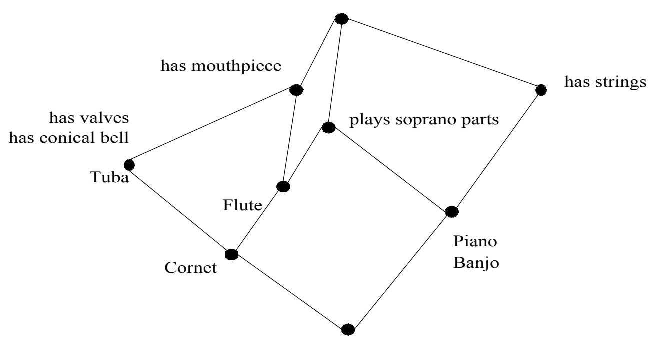


Figure 4. A concept lattice for a context of Musical Instruments.

Beginning with the bottom-most node of the graph, the reader may follow any path to the top-most node. As nodes are traversed along any path, the reader collects a set of elements and attributes shared by those elements. For any node having an attribute label, the nodes at or below that node in the graph form a subset of elements that share that attribute. In general, the higher the node in the graph, the larger the subset of elements it represents, and the smaller the set of common attributes. As a whole, the diagram represents a complete hierarchy of sub-types and instances for a single concept.

For example, consider the line diagram in Figure 4. It depicts similarities among entities, like **Tuba**, **Cornet** and **Flute**, and attributes, like *has valves, has conical bell*, and *has mouthpiece,* with the lines that inter-connect the nodes that represent these entities and attributes. A sub-type label, say Wind-Instrument, might be assigned to this group of instruments which share the attributes of *has valves, has conical bell*, and *has mouthpiece*.

With its origins in lattice theory, FCA provides a visual method for deriving sub-type and super-types among concepts. For this reason, recent research has incorporated FCA with various KA techniques [10]. For more details on the theory of formal concept analysis and concept lattices, please refer to the work in [10], [11], and [12].

2.4 Combined Strengths

The motivation to employ a hybrid is to utilize the strengths of different approaches in overcoming each other's weaknesses. This section summarizes the strengths and weaknesses of conceptual graphs, repertory grids and formal concepts with respect to this work, and shows how we believe the hybrid overcomes some of their weaknesses.

A hybrid approach involving repertory grids promises a technique with a strong foundation in psychology. Grids are employed in surveys and examinations presented to the public. Their popularity suggests that experts can utilize repertory grids with minimal assistance or education about grids. In recognition of their strengths, many KA systems, like AQUINAS, KSS0 and ICONKAT, incorporate repertory grids as a major elicitation technique [5], [13], [14].

As the hybrid approach exploits the strengths of repertory grids in knowledge acquisition, it uses conceptual graphs for their strong capabilities in knowledge representation and inferencing. Conceptual graphs can represent modal and first-order or higher-order logic, with simple and elegant inference rules. For this reason, some researchers suggest conceptual graphs be viewed as a universal modeling language and reasoning tool [15]. A hybrid KA technique stands to inherit these strengths by incorporating conceptual graphs into its design.

Formal concepts have the advantage in that conceptual structures can be generated from the element-attribute set automatically. This makes them self-organizing, relatively free from KE bias, and efficient. Experience with formal concepts suggests that they are easy to read and understand [9, 12] by domain experts.

One strength of repertory grids is that grid values may denote the degree to which attributes apply to specific entities. In conceptual graphs alone, attributes are related to concepts in a Boolean fashion; i.e., per individual, either the attributes are linked to concepts or they are not. On the other hand, conceptual graphs are particularly useful for reasoning and logical inferencing while repertory grids lack capabilities in general reasoning except for heuristic classification. Grid-graphs potentially offer the benefits of both techniques.

A hybrid approach involving conceptual graphs may overcome another limitation of repertory grids. Existing systems that translate repertory grids to knowledge representations have a tendency to treat each construct atomically [16]. For instance, a grid may have two constructs labeled *sounds loud/not loudly* and *sounds softly/not softly* but has no indication that the two constructs are related. A hybrid approach promises to overcome this tendency by associating each grid with a conceptual graph. The appropriate translation of the poles of the constructs to conceptual graphs may identify relationships among constructs. Furthermore, this translation allows the knowledge inherent in each pole and the knowledge inherent to an element to be used for purposes such as inferencing [17].

A hybrid approach permits the multiple techniques to complement and validate each other. We therefore have an avenue for internal validation as concepts and relationships are cast in several forms. If their sense "rings true" during the KA process, we gain confidence that our acquired knowledge is correct. We also gain some generality, since we extend the possible domains to which the hybrid technique may be applied.

3 Troika Approach

The approach used in this paper is outlined in greater detail in [18]. The approach is called **Troika**, after a Russian sled that is pulled by three horses. The three "horses" of the approach are the three theories outlined above: conceptual graphs, (tracked) repertory grids, and formal concept lattices. The essential steps are shown in Figure 5.

Characterize the problem type (e.g., resource-allocation, planning, etc.)
Develop initial conceptual graphs of problem structure
Repeat
Acquire a concept for the knowledge base
Acquire instances of that concept
Acquire a second concept for the knowledge base
Acquire instances of the second concept
Acquire a label for a relation between the two concepts
Acquire knowledge for the relation using a repertory grid
Build a concept lattice from the repertory grid
Acquire any new concepts derived from the lattice
Until no new concepts are acquired.

Figure 5. Major Steps of the Troika Approach.

This approach is only a first step towards integrating these approaches to exploit the power of all three. We are beginning to pursue additional features of these approaches that will further enhance Troika.

3.1 Troika Algorithms

Troika's process is driven by a main procedure (TROIKA) that drives two sub-procedures, an elicitation procedure based on repertory grids (TKE) and an analysis procedure based on formal concept analysis (TKFCA). The TROIKA procedure loops through elicitation and analysis until the user stops or there are no more concepts to find. The TROIKA_ELICITATION procedure uses repertory grids to acquire knowledge which is then analyzed via formal concept analysis in the TROIKA_FCA procedure to help identify sub-types (and hence new concepts). The TROIKA_FCA builds concept lattices and ask the expert to identify by name any relevant sub-concepts identified in the lattices.

The following algorithms form the basis of the Troika technique. Figure 8 shows the elicitation algorithm; Figure 7 shows the subsequent analysis algorithm. Additional work has been done in support of this process; in particular, strategies for determining repertory grid axes along with their corresponding concepts/labels, and various implementation strategies. This additional work is explained more fully in [18].

Step Number	**TROIKA (knowledgebase** *KB**)**
TROIKA-1.	**BOOLEAN KA_done := FALSE;** **while (KA_done = FALSE) do**
TROIKA-2.	*perform knowledge elicitation* **Troika_eliciation (KB);**
TROIKA-3.	*perform analysis of information* **KA_done := Troika_fca (KB);**
TROIKA-4.	**end while;** **End Troika;**

Figure 6. The Main Troika KA algorithm.

Step Number	**TROIKA_FCA (knowledgebase** *KB **)**
TKFCA-1.	***Each repertory grid in the KB represents a concept*** **For each repertory grid, *grid_ExC*, in KB** ***lattice1* := build_lattice(*grid_ExC*)**
TKFCA-2.	**For each node in lattice1** *Each node associates entities and attributes* *Each node identifies a potentially new concept* **concept1 := acquire_concept_from_lattice (lattice1);**
TKFCA-3.	**If (concept1 is not NULL) Then** *The expert has chosen to identify a new concept* **If (exists_in_KB (KB, concept1) = FALSE))Then** **add_concept_to_KB (KB, concept1)** **stop := FALSE;** **Endif;** **Endif;** **Endfor;** **Endfor;**
TFCA-5.	**return stop;** **End TROIKA_FCA;**

Figure 7. The Troika_fca algorithm.

3.2 Example

This section provides some samples of Troika's results. Unfortunately, space does not permit us to show the interactive dialog supported by Troika. Complete examples can be found in [18].

Before Troika starts, a decision is made as to the kind of problem being addressed. In this example, we assume that the domain is some resource-allocation problem. Initial conceptual graphs can be prepared (and re-used) by a knowledge engineer for each kind of domain. The graphs in Figure 9 represent a generic initial set of graphs for resource allocation on which the knowledge acquisition will be based.

Step No.	TROIKA_ELICITATION (knowledgebase *KB)
TKE-1.	*Ask the expert to identify one concept of the domain* ***concept1*** **:= acquire_concept( );**
TKE-2.	**While (*concept1* is not NULL) Do** *the expert has specified a concept to consider*
TKE-3.	**stop := FALSE;** **If (exists_in_KB(KB, concept1) = FALSE) Then add_concept_to_KB (KB, concept1)**
TKE-4.	*Ask he expert to identify the instances of the concept* **If (no catalogue of individuals for concept1) Then** **acquire_catalogue (concept1);** **concept2 := NULL;**
TKE-5.	**While (stop = FALSE) Do**
TKE-6.	**If (concept2 is NULL) Then** *Ask the expert to identify a second concept of the domain* *Any concepts in KB serve as suggestions to the expert for consideration* **concept2 := acquire_concept ( );** **Endif;**

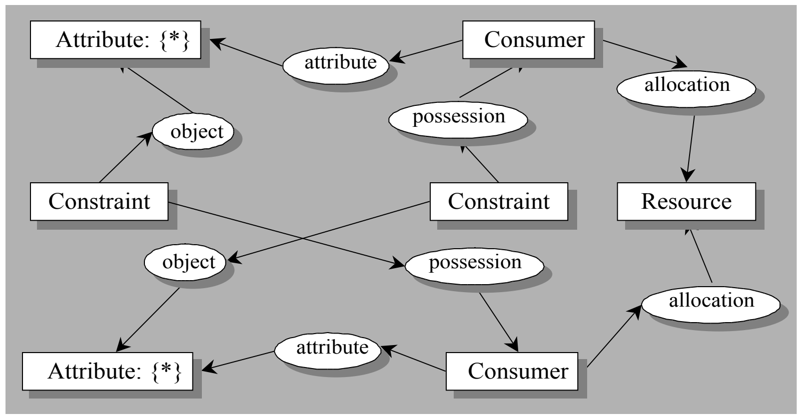
TKE-7.	**If (concept2 is not NULL) Then**
	the expert has chosen to specify a second concept
	If (NOT exists_in_KB(KB, concept2)) Then
	add_concept_to_KB (KB, concept2)

TKE-8.	**If (there is no catalogue of individuals for concept2) Then** *Ask the expert to identify the instances of the second concept* **acquire_catalogue (concept1);** **Endif;**
TKE-9.	*Ask the expert to identify a relation (if any) between the 2 concepts* **relation_1x2 := acquire_relation (concept1, concept2)** *;*

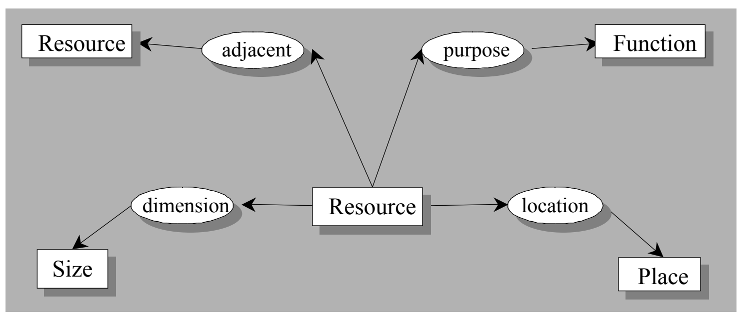
TKE-10.	**If (relation_1x2 is not NULL) Then**
	the expert has chosen to specify a relation between the concepts
	Ask the expert if relation_1x2 should be removed from KB
	If (response is YES) Then
	Remove the unwanted relation from the KB
	KB := remove_relation_from_KB (KB, relation_1x2);
	Else
	Display the repertory grid to acquire information
	grid_1x2 := build_grid (concept1, concept2, relation_1x2) ;
	Display the grid, grid_1x2 and ask the expert to fill grid
	acquire_grid_values (grid_1x2);
	KB := add_grid_to_KB (KB, grid_1x2);
	Endif;
	Endif; *IF (relation_1x2 is not null)*
	concept2 := null;

TKE-11.	**Else** **stop := True;** **Endif;** *IF (concept2 is specified)* **End while;** *WHILE (stop = FALSE)*
TKE-12.	*Ask the expert to identify a (new or existing) concept to consider* **concept1 := acquire_concept ( );** **End while;** *WHILE (concept1 not null)* **END_TROIKA_ELICITATION;**

Figure 8. The Troika_elicitation algorithm.



(a)



(b)

Figure 9. Set of initial conceptual graphs for KA.

We have applied Troika to the room allocation problem of Sisyphus-I. We summarize here the Troika steps with respect to acquiring knowledge about that domain. There are various constraints placed on the allocation, such as: the head of a group should be in a central office, the secretaries should be close to the head of the group, smokers and non-smokers should not share an office, etc. The KA process should therefore center on rooms as the resource, and people as the consumers of that resource; in the Troika process, such a correspondence is acquired naturally through an interactive dialog.

For each concept specified, the expert must specify some instances of that concept. For example, when the expert specifies the concept **Size**, she subsequently specifies the set of instances {Large, Medium, Small}. These instance names are used to produce the construct labels and element labels for the repertory grids in the grid-graph. In the Sisyphus-I example, the expert specifies (through a dialog, not shown here) nine relations or pairs of concepts, for the grid-graph during the initial iteration of the Troika algorithm. A summary of these nine relations is shown in Figure 10. The expert then populates these grids with values.

Concept #1 Label	**Concept #2 Label**	**Relation Label**	**Repertory Grid Title**
Person	Room	assignment	What Room is an assignment of Person?
Person	Attribute	attribute	What Attribute is an attribute of Person?
Room	Size	dimension	What Size is a dimension of Room?
Room	Function	purpose	What Function is a purpose of Room?
Room	Place	location	What Place is a location of Room?
Room	Room	adjacent	What Room is an adjacent of Room?
Person	Attribute	attribute	What Attribute is an attribute of Person?
Person	Constraint	possession	What Constraint is a possession of Person?
Constraint	Attribute	object	What Attribute is an object of Constraint?

Figure 10. Repertory grid titles for room allocation problem.

The repertory grid titled "What constraint is a possession of person?" is shown in Figure 11. We assume that these entries were acquired from domain experts through the TROIKA_ELICITATION procedure. The formal concept lattice derived by TROIKA_FCA is shown in Figure 12. This allows us to determine new concepts (as sub-types) as shown in Figure 13.

Sam	**Ira**	**Sue**	**Bob**	**Louise**	**Juan**	**Preetha**	**Marjorie**	
-	-	-	+	-	-	+	+	*Needs "must have office to self"*
-	+	-	-	+	-	-	-	*Needs "officemate works on Project A"*
-	-	+	-	-	+	-	-	*Needs "officemate works on Project B"*
+	-	-	-	-	-	-	-	*Needs "officemate does clerical work"*
+	+	+	-	+	+	-	-	*Needs "officemate is not a manager"*
+	+	+	-	+	+	-	-	*Needs "prefers non-smoking officemate"*
-	-	-	+	-	-	+	+	*Needs "room near the secretary"*
+	-	-	-	-	-	-	-	*Needs "room to be centrally located"*

Figure 11. Repertory grid: What Constraint is a possession of Person?

Troika's operation uses a natural language dialog, with the terminology closely paralleling that of a typical repertory grid process. Again, space does not permit us to show the dialog here. Instead, we show the resulting conceptual graphs, with the repertory grids embedded in them. In **Figure 14**, we show the grid-graphs as enhanced through the acquisition from. The original graphs from Figure 9 are shown in **bold** to emphasize the additional knowledge acquired. Note that each repertory grid concept's literal referent (i.e., the underlying grid) contains valuable knowledge about individuals comprising the extent of the concepts in this graph. Note also that a number of sub-types (not shown) have been identified through formal concept analysis.

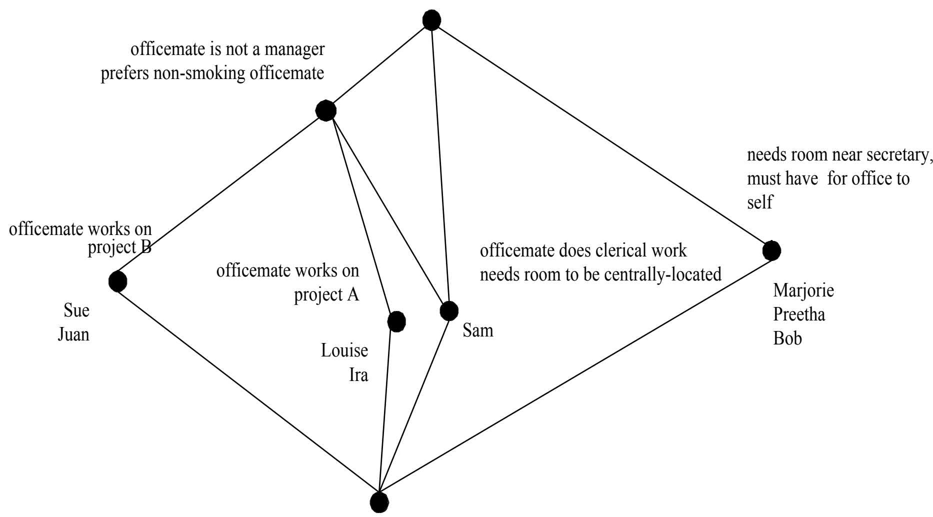


Figure 12. Lattice derived from the Person(X)Constraint grid.

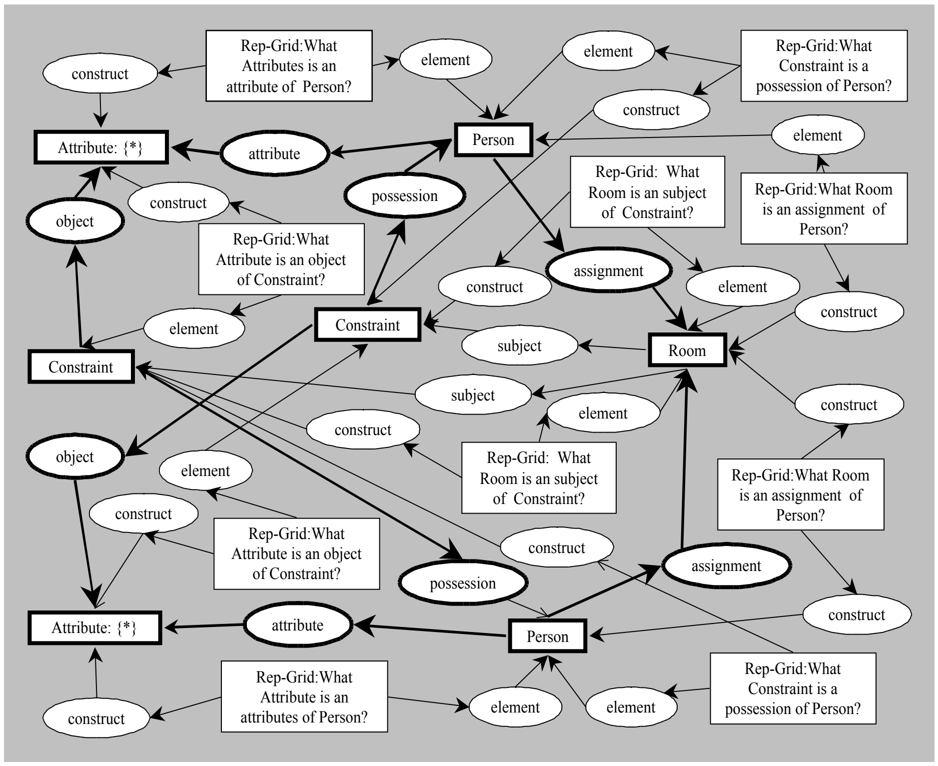
Sub-type Name	**Entities**	**Attributes**
Worker	Sam, Sue, Preetha, Juan, Ira, Louise, Bob, Marjorie	(prefers non-smoking officemate) and (officemate is not a manager)
Secretary	Sam	(needs room to be centrally-located) and (officemate does clerical work)
Project A programmer	Sue, Juan	(officemate works on project B)
Project B programmer	Ira, Louise	(officemate works on project A)
Manager	Bob,Preetha, Marjorie	(must have room to self) and (needs room near secretary) and (needs room to be near secretary)

Figure 13. Sub-types derived from grid Person(X)Constraint.

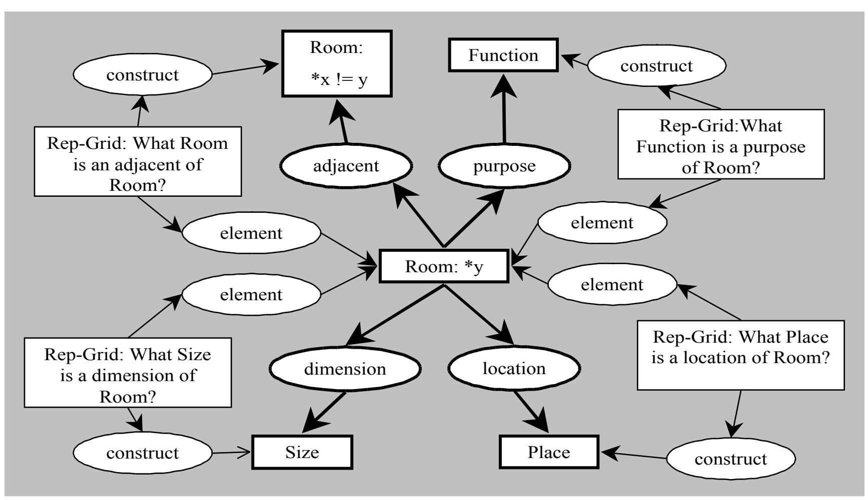
4 Conclusion

In this paper, we have illustrated that flavor of the Troika approach. The work in [18] shows Troika also being applied to the assembly of the International Space Station (ISS). One notable (and encouraging) finding is that although in both domains we started with the identical set of graphs, the resulting conceptual graphs of the ISS-Assembly problem are quite different than the ones shown here. We therefore have some evidence that the initial graphs do not appear to overly constrain the KA process and allow KA to proceed in a natural way. This provides some sign that the technique has some generality which we will validating in future work.

We are also encouraged by the fact that we are using only the basic features of these three approaches, yet a great deal of power is already evident. This lends some support to the approach and gives us some evidence that incorporating additional features from the approaches will make Troika even more useful.



(a)



(b)

Figure 14. Grid-graphs resulting from TROIKA.

5 References

1. Diaper, D., Knowledge Elicitation: Principles, Techniques and Applications. 1988, New York: Halsted Press.
2. Sowa, J.F., Knowledge Representation: Logical, Philosophical, and Computational Foundations. 2000: Brooks/Cole.
3. Tepfenhart, W. and W. Cyre. Conceptual Structures: Standards and Practices. in 7th Intl. Conf. Conceptual Structures. 1999. Blacksburg, VA, USA: Springer-Verlag.
4. Kelly, G.A., Psychology of Personal Constructs. 1955, New York: W.W. Norton.
5. Bradshaw, J.M., et al., How to Do With Grids What People Say You Can't. Journal of Knowledge Acquisition for Knowledge-Based Systems, 1987. **1**(4).
6. Gaines, B.R. and M.L.G. Shaw, Knowledge Acquisition Tools Based on Personal Construct Psychology. Knowledge Engr. Review, 1993. **8**(1): p. 49-85.
7. Wolf, R.P. and H.S. Delugach, Knowledge Acquisition via Tracked Repertory Grids, . 1996, Computer Science Dept., Univ. Alabama in Huntsville.
8. Delugach, H.S., Repertory Grid Graphs: A Hybrid of Conceptual Graphs and Repertory Grids, in Proc. 11th Knowledge Acquisition Workshop (KAW-98), B. Gaines and M. Musen, Editors. 1998, University of Calgary: Banff, Alberta, Canada.
9. Spangenberg, N. and K.E. Wolff. Formal concept analysis of repertory grids: a case study of a patient with anorexia nervosa. in Proc. 17th Eur. Conf. on Psychosomatic Res. 1988.
10. Erdmann, M. Formal Concept Analysis to Learn from the Sisyphus-III Material. in Knowledge Acquisition Workshop. 1998. University of Calgary.
11. Wille, R., Concept Lattices and Conceptual Knowledge Systems. Computers Math. Applic., 1992. **23**(6-9): p. 493-515.
12. Ganter, B. and R. Wille, Formal Concept Analysis: Mathematical Foundations. 1999, Heidelberg: Springer-Verlag.
13. Shaw, M.L.G.a.G., B.R., KITTEN: Knowledge Initiation and Transfer Tools for Experts and Novices. Intl. Jour. of Man-Machine Studies, 1987. **27**(3): p. 251-280.
14. Ford, K.M., et al., ICONKAT: An Integrated Constructivist Knowledge Acquisition Tool. Knowledge Acquisition, 1990. **3**(2): p. 215-236.
15. Sowa, J.F., Conceptual Graphs Summary, in Conceptual Structures: Current Research and Practice, P. Eklund, et al., Editors. 1992, Ellis Horwood. p. 3-52.
16. Bringmann, M.W. and F.E. Petry, A Semantic Network Representation of Personal Construct Systems. IEEE Transactions on Systems, Man and Cybernetics, 1992. **22**(5): p. 1161-1168.
17. Wolf, R.P., Knowledge Acquisition via the Integration of Personal Constructs and Conceptual Graphs, in Computer Science Department, Ph.D. dissertation. 1998, University of Alabama in Huntsville: Huntsville, AL, USA.
18. Lampkin, B.E., Troika: A Hybrid Approach To Knowledge Acquisition And Representation Using Graphs, Grids And Lattices, in Computer Science. 1999, Univ. of Alabama in Huntsville: Huntsville, AL, USA.