	Protégé-2000	Arezzo / PRO*forma*	Design-a-Trial
General Information			
A. Purpose	Protégé-2000 is a general-purpose knowledge acquisition environment that can be used to build guideline models, and encode clinical guidelines using those models.	Arezzo allows encoding guidelines using the PROforma guideline model. Using the AREZZO, new clinical applications can be quickly modelled and tested, and instantly deployed on the Internet.	Design-a-Trial (DaT) is a knowledge-based decision support system for authoring clinical trial protocols.
B. Target Users	Developers who build guideline models and domain specialists who enter guideline knowledge.	Domain specialists and developers who enter guideline knowledge	Physicians who are involved in designing clinical trials. This includes (targets) clinicians who are less experienced in RCT (randomized controlled trial) design.
C. Institution / people – Who are the developers of the workbench?	Protégé Group Stanford Medical Informatics Stanford University School of Medicine (Mark Musen, Ray Fergerson, Natasha Noy, Jennifer Vendetti, Monica Cubrezy, …)	Arezzo is based on the PRO*forma* language, developed at the Imperial Cancer Research Fund's Advanced Computation Laboratory, UK. It is one of the products developed by Infermed Ltd., UK	S Modgil, P Hammond --- Biomedical Informatics Unit (Eastman Dental Institute for Oral Health Care Sciences); JC Wyatt, H Potts --- Knowledge Management Centre (School of Public Policy) University College London
D. Time frame – When did the project start?	The original Protégé was built in 1988 as part of Mark Musen's PhD thesis. In the past 14 years, it has gone through 4 distinct releases to the current system, Protégé-2000.		DaT 1.1 was developed in the early 1994. This was updated from OS/2 to Window NT base with DaT 1.2 which was completed in 1998.

E. Status – Is the project completed, ongoing…? Is the software a demo, a research prototype, commercial …?	The project is ongoing. Protégé-2000 is high-quality research software, used by hundreds of academic and industry groups.	Arezzo is a commercial product. Research at the Cancer Institute is ongoing.	DaT 1.1 and 1.2 are prototype versions. Appears this has mainly undergone some preliminary evaluations. Preliminary work is ongoing for DaT 2.0. Current work aims at a comercial product to be release around 2003. ("UCL, through the Eastman and the School of Public Policy, will be working with InferMed Ltd on a new £212,000 Teaching Company Scheme programme to develop a commercial version of clinical trial design software." InferMed Ltd: http://www.infermed.com/)
F. Availability – For those outside the project, are the workbench software and models freely downloadable, available under license, unavailable…?	Protégé-2000 is freely downloadable from http://protege.stanford.edu website under an open source license.	Arezzo is a proprietery product developed by Infermed Ltd. You need special permission to download Arezzo for use in research labs.	Not that we were able to confirm. Commercial version may be available in 2003.
G. Applications – How and where is the workbench being tested or used?	Protégé-2000 is a general-purpose domain-independent knowledge-acquisition tool. It has been used by groups in varied fields, inside and outside medical informatics. It is extensively used by three guideline modeling groups: EON , Prodigy and InterMed.	Arezzo is being used in a wide range of guideline-based applications such as applications that faciliate early referrals decision support for HIV and a number of other guidelines, andgenetic risk assessment.	Mainly by those envolved in the current project (the UCL group).
H. Installed base and numbers of users currently employing the software; the purpose of their use.	Protégé-2000 has an active user community that includes research and industrial projects in more than 100 countries. There are about 3,500 registered users. About 50 groups have provided descriptions of their projects. This list includes not only projects which are actually using Protege but also projects which have or are currently evaluating Protégé as well as even some "competitors" to Protégé.	PROforma has been used to develop ERA, a set of 10 cancer guidelines for early refererals in cancer currently being evaluated in association with the UK NHS Information Authority. PROforma technology for authoring and publishing executable clinical guidelines is being commercialised (under the Arezzo brand name) by InferMed Ltd. in London.	Not readily available.
Components			

I. Guideline model – What is the underlying guideline model? Is the guideline model geared towards any specific types of guidelines?	Protégé-2000 is not tied to any guideline model. It can support relatively simple guideline models such as Prodigy to complex guideline models such as EON.	Arezzo is a dedicated authoring tool to create computable guidelines based on the PROforma guideline model.	The ontology is written using Protégé-2000 version 1.3.4. This is augmented by a large Prolog rule-base. The current prototype has a knowledge base of thoracic medicine only. (These authors have published work with Oncology knowledge base too.)
J. What are the capabilities supporting, or supporting development of, the following guideline features:			
a. Enterprise workflow context and modeling	Samson has developed workflow models (in collaboration with University of Pavia in Protégé-2000. However, these models have not been implemented.	The PROforma method has been applied to workflow managers in the treatment of cancer, asthma and other diseases. It is not clear if there is any special support to model workflow	Not applicable to the clinical trial use-case.
b. Information processing context and modeling	No experience in modeling system resources.	None.	Appears to assume a custom user interface for this relatively stand-alone task.
c. Graphical (flowchart logic) depiction	One of the special-purpose widgets called the Diagram Widget allows users to model flowcharts. This widget has been used to model clinical algorithms.	Arezzo models a guideline as a series of tasks that are networked together. The nodes represent tasks and the arrows among them represent the sequencing order. The Composer module (similar to Protégé-2000's Diagrm widget) allows users to easily build these networks. It also provides appropriate GUI forms to enter relevant information on each task.	Employs a simple graphical representation of the components of a trial emphasising the typical order in which the main design subtasks should be undertaken.

d. Data layer instantiation of logical elements into standard data elements	In the EON project, patient data variables were defined in the guideline model and subsequently mapped to data elements in a relational database.	The enquiry task can be implemented to request patient information from the user, retrieve information from a database, or extract features from an image.	There is a concerted effort to align the purpose specific clinical trials ontology with the GLIF ontology and related work undertaken among Protégé users.
e. Execution engine for run-time support?	Since Protégé-200 can support different guideline models, there is no generic execution engine. It provides a rich set of API to access the elements in the knowledge base. An execution engine needs to be built for each guideline model.	Arezzo has an execution engine that executes a guideline treatment plan by interpreting the tasks in a specific sequence using patient data. Arezzo provides a GUI that shows an overiew of task execution, task state, recommendations, and any enquiries for information.	Not applicable to the clinical trial use-case.
K. EMR – What is the model of patient information?	Protégé-2000 is a generic knowledge acquisition tool. It does not come with a built-in access to a virtual EMR. However, the functionality can be added as part of the guideline models.	There is no explicit patient information model.	No EMR
L. Controlled Terminology Services – Does it provide access to controlled terminology services? How smooth is it to use standard terminologies when entering guideline knowledge? Are there utilities for loading and maintaining versions of external terminologies?	The mode of operation to use external terminologies is to import the whole ontology into the Protégé-2000 environment. Protégé-2000's component-based open architecture facilitates integrating utility functions and custom-built applications into the system. For example, the developers can add new functional tabs to the standard set. At knowledge acquisition time, users can access the utility functions via the new tabs. One functional tab that is relevant to encoding guidelines allows online access to UMLS. The UMLS tab allows users to browse online sources, to verify existence of a medical concept within UMLS, and to import sub-trees of the UMLS ontology directly into the knowledge acquisition environment. Apelon recently built a Protégé tab that provides access to the Distributed Terminology Service (DTS). Protégé-2000 does not provide any utilities for maintaining terminology versions. No guideline group has yet used terminology services within Protégé-2000 when entering guideline knowledge.	No support for controlled terminology service.	The ontology is written using Protégé-2000 version 1.3.4. DaT 1.2 : knowledge base, encoded in Prolog, is divided into a collection of medical facts (currently limited to thoracic medicine) and a set of expert rules. These expert rules have been encoded as constraints, and are used to generate critiques of methodological, medical, statistical and ethical aspects of clinical trial design. The knowledge base also contains definite clause grammars (dcgs) used in the generation of the protocol and critique texts. Ontologies provide a means for structuring data, and can assist in specification of a system, helping to identify system requirements and to understand relationships among system com-ponents. They have developed an ontology for RCTs, and are currently coupling design of the DaT 2.0 interface with the ontology. They are also defining a mapping between the ontology and the Prolog symbolic schema that is instantiated by data entered when designing a trial. In this way, changes to the ontology can readily be integrated into the underlying symbolic representatio

Modeling & Encoding Process			
M. Mode of Operation – What is the general process to encode guidelines? Does it support multi-layered modeling that allows clinical experts to interact easily with knowledge experts?	The developer with the help of domain specialist creates an ontology of domain concepts, and builds a patient model and a guideline model using these concepts. The domain specialist with the help of the developer enters guideline knowledge using user-interface forms and special-purpose widgets. Currently there is no explicit support for multi-layered modeling but it can be achieved to an extent through special-purpose widgets and conventions on division of labor between clinical experts and knowledge experts.	The developer assembles the required medical domain concepts. Then the domain specialist uses the Arezzo Composer to lay out a task network for the application, sketching the tasks that are required and any scheduling constraints on their execution. When the designer is satisfied with the layout of tasks thier detailed definitions are added, indicating general properties like the timing or cycling attributes of a task, and specific task properties such as recommendation rules for decisions. Thus it does allow multi-layered modeling. Using a verification tool, the logical operation of the system is validated. It can then be equipped with a suitable user interface or embedded in a host applicaiton.	Design-a-Trial interviews a physician, prompts and guides them through suitable design options, comments on the statistical rigour and feasibility of their proposed design, and generates a 6-page structured draft protocol document. There is no direct communication between clinical trial authoring users and domain experts. Specifically, the Prolog rule-base authoring appears distinct and disconnected from the target user base.
N. Multi-user support – What kind of multi-user support does it provide? Does a client software allow multiple remote users to work collaboratively?	When using a database backend, multiple Protégé users can work on the same knowledge base at the same time. Currently users are limited to working on different parts of the knowledge base. There is ongoing effort to build a client-server architecture to improve on the functionality. There will be an indication when a new user starts to edit the knowledge base, and a user's changes to the knowledge base will be propagated to other users.	There is no multi-user support.	There is no mention of multi-authoring capability, nor of tracking and merging resources that would be so required.
O. Extensibility – How extensible is the system? Does it have a library of components that can be assembled in different ways? Does the database or programming environment create any known restraints of scale?	Protégé-2000 has an open-source Java-based extensible architecture that allows developers to build special-purpose GUI widgets and utility functions that can be easily integrated with the core system. Protégé-2000 has been used to build decision-support systems based on guideline models that embody very different assumptions, such as EON and Prodigy. EON is very expressive and uses complex constructs such as PAL constraints and temporal abstractions to represent complex decision-criteria and patient states. Prodigy is a simpler model that stresses being intuitive to domain-specialists, and relies more on clinicians to recognize complex clinical patterns at the time of guideline execution. Protégé-2000's plug-and-play framework allowed both the modeling groups to customize the knowledge acquisition environment to suit their models.	Arezzo's sole pupose is to provide an environment to facilitate guideline modeling using the PROforma model. It provides a rich set of tools to do just that. Arezzo has been used to build a wide range of clinical decision-support applications. It does not support the kind of extensions you can make with Protege-2000.	The system invokes an underlying clinical trials ontology, authored in Protégé. Thus, there is an implicit extensibility of this knowledge. Similarly, the action contraint rule-base is managed in Prolog, which also has intrinsic abilities to scale information. However, there does not appear to be explicit modularity or data re-use within the clinical trial authoring environment per se, specifically there appears to be no obvious mechanism for "sub-routining" clinical trial protocol components or re-using trial elements in any way.

I. User-friendliness – How does it make it easy for domain experts to enter guideline knowledge? How well does it hide the complexities of the underlying guideline model? What visual metaphors does it use to aid the knowledge entry process? Are the component modes of operation understandable, scalable and useful for: **a. the clinical domain expert** **b. the knowledge engineer** **c. the software maintenance vendor?**	Protégé-2000 generates user-interface forms automatically based on class definitions. Users build knowledge bases by filling out the form. Besides these generic forms, special-purpose user-interfaces can be integrated to facilitate knowledge acquisition. For example, a diagram widget that presents information graphically as a network of nodes and arcs has been successfully used to encode clinical guideline algorithms. Such widgets can be effectively used to also hide the complexities of the underlying guideline model.	Arezzo composer provides a user-friendly evironment to build the network of tasks using the different PROforma constructs. It has a diagramming tool that allows assembling of tasks as a network of nodes. It also has a knowledge editor to specify the details of a task, an decision editor to enter decision rules, and a condition editor to define a wide range of logical conditions that may be relevant during task execution. The user-interfaces are special-purpose and greatly simplify the modeling process.	The DaT 1.2 interface, implemented in Prolog, employs a simple graphical representation of the components of a trial emphasising the typical order in which the main design subtasks should be undertaken. The user is presented several data entry forms. The Prolog sophisticated graphical user interface was felt to be impractical to implement. Therefore, with DaT 2.0, uses a visual basic interface and an Amzi Prolog logic server module. The environment was designed for a naive clinical user, and would have less utility for domain experts or knowledge engineers.
P. Evidence – When entering guideline rules, is there a way to specify the references to medical literature and/or enterprise standards of care that justify the rules?	Yes. In EON, the guideline model had place holders associated with the knowledge rules to specify references that justify the rules.	One of the pillars of the PROforma approach is providing argumentation for a specific recommendation. Therefore, when entering recommendations, designer can specify evidence for and against such as recommendation. It is not clear whether links to literature can be specified at that time, and the clinician will have access to the appropriate links when executing the guideline.	There is no evident mechanism in the description, though the the use case of clinical trial authoring mitigates this requirement.
Q. Does the software support maintenance of multiple versions with rollback and compare functionality?	No.	No.	There is no mention of these features.
Verification, Simulation & Localization			
R. Verification – What are the mechanisms to verify the guideline knowledge base? Internal scenario data integrity and consistency? Compliance with external vocabulary standards? Compliance with syntax standards for logic expression?	Protégé-2000 supports a constraint language called PAL which can be used to write complex integrity constraints on the knowledge base. PAL allows developers to make general assertions about relationships among objects in Protégé-2000 (e.g., “all criteria instances are referenced, “nodes” in a diagram should be connected to other nodes”), and to check if these relationships hold directly in the knowledge base.	Arezzo tools generate a definition of the application knowledge base into R2L, a declarative language. With the formal model of the general properties of decision, plans and many of the constraints within and between tasks it is possible to automatically identify problems or potential problems in an R2L specification. Arezzo can detect any incorrect datatypes, invalid syntax of attribute values, critical missing values, inconsistent scheduling constraints, etc. It can generate a report of errors that would prevent the application from executing, and warnings about properties that do not prevent execution but may suggest omissions or similar errors.	The major use-case is to validate that clinical trials developed in DaT are consistent or well-formed with respect to an internal library of Prolog knowledge rules. Exception over-rides are allowed, but require explanation. There is no effort to enforce the use of controlled terminologies in the protocols developed using the tool.

S. Simulation – Does it provide support for guideline simulation so that new guideline knowledge can be rapidly tested?	End-user applications that take the ontologies and the knowledge base as input can be plugged in as tabs just like utility functions. Since changes in the knowledge base are immediately available to the application, they can be tested rapidly using the application tab. This facility was effectively used in the ATHENA project when building a hypertension advisory system. Using the application tab, domain experts could rapidly test the advisory system and the entered hypertension guideline knowledge base. They could modify parts of the knowledge base, and immediately see the effects of their changes in the advisories generated by the application. They could also verify the knowledge base against different patient data.	Arezzo's Protcol Tester is part of the development environment. It has an execution engine that is able to execute tasks by carrying out actions or finding out the current state of the environment by making requests to a human user or software system (such as a database). It displays a set of decision options that it is recommending, arguments for each option. Thus a complete guideline can be executed within the development environment facilitating rapid testing of the system.	There was no mention of this feature.
T. Localization – What kind of support does it provide for localizing a generic version of an encoded-guideline for particular institutions?	There is no explicit support for localization.	There is no explicit support for localization.	The language and specifications for clincal trials protocols developed using the tools are not constrained in any way, implying virtually complete abilities to localize a protocol. However, there appears to be no mechanism to migrate this protocol to another "location."
SUMMARY			
U. What are its strengths?	• Protégé-2000's extensible component-based architecture and configurable GUI greatly facilitates customizing knowledge-acquisition for given domains. • Automatic generation of domain-specific user-interface forms cuts down on the time and effort needed to go from building knowledge models to acquiring knowledge via the models. It exposes the guideline model to the domain-specialists immediately. This rapid turnaround can be vital to guideline model evolution and experimentation. • Custom user-interface widgets such as the Diagram widgets can be integrated to ease knowledge acquisition of complex information. • Utility functions such as terminology services, and end-user applications can be plugged-in easily to expand the support for knowledge acquisition. • PAL constraint language is expressive and can be used to write complex integrity constraints on the knowledge base. Another use of PAL is in writing decision-criteria which define patient-specific constraints that must be evaluated during guideline execution. • The organization of knowledge bases as projects, and the notion	• Arezzo is commercial product that probably has gone through the rigor of commercial software development process. • Arezzo is tightly coupled with the PROforma guideline model. It has elegant and highly focussed GUI that provides excellent support for the modeling process. • The integration of the execution engine with the guideline encoding environment allows rapid testing of the knowledge base and the application itself. • Arezzo provides a rich API that facilitates the technology to be embedded in larger clinical applications. • It provides a strong and explicit support for providing explanations for its recommendations. • It uses a concise language to define conditions and temporal constraints, and has an editor that makes it easy to enter expressions.	• Updated version (DaT 2.0) planned to have a more user friendly interface using visual basic. • It is built upon a well-formed ontology about randomized clinical trials, developed using Protégé. • It has a large library of knowledge rules, written in Prolog • It appears to have well-formed internal constraint rules guiding the authorship of trails. • It appears to enforce accepted standards of trial power and statistical relevency.

V. What are its weaknesses?	• Special-purpose knowledge acquisition tools such as Arezzo and AsbruView are tightly coupled with the underlying guideline model. Such tools generally provide elegant and sophisticated user-interfaces that are highly directed. Protégé-2000 provides generic user-interface forms that may not be intuitive to use for a domain-specialist. For example, Protégé-2000 associates one form with each class and does not facilitate logical grouping of classes into a single form. Therefore, it provides a general forms-based view of guideline knowledge in a knowledge base, but not a concise and domain-specific view. Domain specialists can find it daunting to review the entered knowledge form by form. • There are no 'wizards' to guide the domain-specialist through the knowledge acquisition process. Thus knowledge-entry can be unstructured, and fragmented. Domain specialists may lack the sense of how to go about entering knowledge, what they have entered so far and what needs to be entered. • There has been no demonstration of how a standard terminology service would be integrated with Protégé-2000, and used in the gui • During encoding a guideline, domain specialists need to be able t	• As in Protégé, there are no 'wizards' to guide the domain-specialist through the knowledge acquisition process. • There is no support for controlled terminology services • There is no facility to generate a paper document of the encoded guideline. • The language used for expressing conditions can be limited. • It is not clear how you reuse domain concepts. • There is no facilitty to include one guideline as part of another guideline.	•Updated version (DaT 2.0) is not available until 2003. •Knowledge base appears to be preloaded….therefore, we suspect adding new knowledge bases may be difficult for the end user. •Software has not been tested widely, best we can tell. •There is no effort to direct resulting trial protocols to invoke health data standards or controlled terminologies.
W. References	Shankar RD, Tu SW Musen MA. Use of Protégé-2000 to Encode Clinical Guidelines. Proc. of the AMIA Annual Symposium, 2002; (submitted). Musen MA, Fergerson RW, Grosso WE, et al.. Component-Based Support for Building Knowledge-Acquisition Systems. Conference on Intelligent Information Processing (IIP 2000) of the International Federation for Information Processing World Computer Congress (WCC 2000). Beijing, 2000:18-22. Protégé-2000 documentation at http://protege.stanford.edu	Fox J. and Das S. Safe and Sound: Artificial Intelligence in Hazardous Applications. AAAI, Menlo Park, CA, and MIT press., Cambrisge, MA, 2000. http://www.infermed.com SMI has license for Arezzo Composer	Wyatt J, Altman D, Heathfield H, Pantin C (1994): Development of Design-a-Trial, a knowledge-based critiquing system for authors of clinical trial protocols. Comp Prog Meth Biomed, 43, 283-291.

	GLIF Guideline Authoring Tool	GUIDE	AsbruView
General Information		GUIDE exists as a portion of a patient care flow system and research project termed PatMan. GUIDE is the graphical interface for drawing an algorithm representing the guideline. The ouput of guide is utilized in various ways.	
A. Purpose	GLIF Guideline Authoring Tool is a workbench designed to enable encoding of clinical guidelines in the GLIF3 format.	Guide is part of a patient centered workflow system called PatMan. Guide is the graphical front end that supports the acquisition of gl knowledge and converts it to a petri net based workflow representation.	AsbruView is a software user interface that provides visualization (and some editting) of guidelines/plans written in the Asbru guideline representation language.
B. Target Users	The GLIF3 methodology is to have clinicians encode a top-level cnceptual view of a guideline and have knowledge engineer encode the computable parts. The GLIF workb bench does not yet support this distinction.	Domain experts and developers at (institution). The overall system includes modules for content specialists, knowledge engineers, clinical use and administration.	Non -technical physicians (after some training), who need to visualize Asbru guidelines.
C. Institution / people – Who are the developers of the workbench?	The Decision System Grouop at Brigham and Women's Hospital, Harvard Medical School developed the tool.	The department of Informatics and Systems at the University of Pavia, Italy. The developers include Silvana Quaglin and Mario Stefanelli.	Asgaard Project, Institute of Software Technology, Vienna University of Technology, Vienna, Austria, (Robert Kosara, Silvia Miksch).
D. Time frame – When did the project start?	The Intermed project started in July of 1999.	Guide description published in 1998. Project seems to have been going on since 1995. Most recent publications on Patient Care Workflow system in 2001.	I presume that AsbruView started in the mid-1990's, as part of Kosara's MS thesis.

E. Status – Is the project completed, ongoing…? Is the software a demo, a research prototype, commercial …?	The Intermed project will end in December of 2002. The sofware is a research prototype.	Software appears to be a research prototype with one or two example projects. There is a re-write currently in progress (according to Samson). A demo film states there are 10 projects and 1million pounds coming into the lab.	Info unavailable - pending email response from Kosara.
F. Availability – For those outside the project, are the workbench software and models freely downloadable, available under license, unavailable…?	As of July 2002, it is not available outside the Intermed project	A java based demo of GUIDE is available for download. It is slow and ? Reliability. There are Lotus screen cam based demos of the workflow system also available on the web site http://aim.unipv.it/projects/patman There is also an available ontology editor (webont) and a query system at enrich.open.ac.uk/patman	Info unavailable - pending email response from Kosara.
G. Applications – How and where is the workbench being tested or used?	The workbench has being tested by researchers at Intermed's collaborating sites at DSG, Stanford Medical Informatics and Columbia.	Tested with a guideline for the management of acute myeloid leukemia in children (1). The have also modeled the operations of a stroke unit utilizing the AHA Stroke Guidelines (3). There appears to be a uk web site devoted to discussion of using PatMan with pressure ulcers. There is a browsable ontology there.	AsbruView is a special-purpose tool, designed for visualization and editting of Asbru guidelines. Unknown if it is used outside the Vienna University of Technology.
H. Installed base and numbers of users currently employing the software; the purpose of their use.	The software has not been released outside the project.	Unknown. One of their Lotus Screen Cam demonstrations mentions 10 investigational projects and 1 million pounds in income at their lab.	Unknown if it is used outside the Vienna University of Technology.
Components			

I. Guideline model – What is the underlying guideline model? Is the guideline model geared towards any specific types of guidelines?	GLIF3 is the underlying guideline model. The current workbench is mostly a tool for clinicians	The underlying model is based on the use of PetriNets and Relational tables. The system is implemented in an Oracle Workflow engine. Guide utilizes a representation similar to Protégé to draw the guideline.	AsbruView was designed and built specifically for visualization of guidelines represented in the Asbru language. Asbru is a "plan-representation" language that uses LISP-like syntax tp represent clinical guidelines as time-oriented skeletal plans.
J. What are the capabilities supporting, or supporting development of, the following guideline features:			
a. Enterprise workflow context and modeling	None	PatMan is billed as a Patient Careflow System. GUIDE is the graphical front end to create PetriNet based clinical workflow models. To that end an Enterprise Ontology is incorporated into the system. The Enterprise Ontology is maintained at the Standford Knowledge Systems Lab.	AsbruView does not integrate any specific workflow model, but couldprobably be used to model a variety of clinical or enterprise workflows.
b. Information processing context and modeling	None	The user of GUIDE models the guideline flow utilizing an algorithm based model similar to the protégé interface.	AsbruView does not appear to be designed to represt the information processing context or local resources required for guideline implementation.
c. Graphical (flowchart logic) depiction	The major feature of the workbench is a graphical tool for creating flowchart. It automatically lays out a flowchart.	The algorithm representation is similar to the graphial representation in Protégé. Guide is written in Java.	The primary purpose of AsbruView is visualization of guideline flow for physicians. The AsbruView user interface presents two views concurrently: (1) A "**Topological View**" that displays relationships among plans (i.e., sub-plans within guidelines), and (2) A "**Temporal View**" that displays the temporal characteristics of plans in more details. In the Topological View, plans are depicted as segments on a visual "running track" metaphor. It is important to note that the topological view metaphor is from the point of view of the clinician (not the patient) moving along a running track populated with guideline sub plans. Selected process characteristics (e.g. entry, exit conditions) are also displayed using "traffic" metaphors such as stop-lights and entry gates. The temporal view employs less intuitive symbols (reminiscent of a Gantt chart), to show detailed temporal relations within and among plans.

d. Data layer instantiation of logical elements into standard data elements	None	PatMan contains an organizational model and is supposed to support the mapping of guideline steps to actual organizational resouces including the EMR.	??
e. Execution engine for run-time support?	None	The guideline is translated into Petri Nets for analysis and simulation. The Petri nets are then imported as workflow representation that can run in the Oracle Workflow environment.	While Asbru's LISP-like syntax seems to imply that it is intended to be computable, the authors specifically state that the aim of Asbru is to support "design and execution of skeletal plans . . . ***by a human executing agent***".
K. EMR – What is the model of patient information?	The GLIF3 model uses a set set of the HL7 RIM classes as the model of patient information. However, the workbench does not support that at this time.	There is not an integrated EMR. The Oracle Relational Model can be extended to integrate clinical data. The necessary extensions to the workflow representation for clinical use are an area of investigation. They utilize a enterprise ontology developed external to their site.	AsbruView has no inherent patient information model. Asbru (the language) appears to represent skeletal plans only (i.e., plans that do not contain or interact with patient data).
L. Controlled Terminology Services – Does it provide access to controlled terminology services? How smooth is it to use standard terminologies when entering guideline knowledge? Are there utilities for loading and maintaining versions of external terminologies?	The GLIF3 model requires that the terms used in describing patients be selected from a terminology. The workbench does not suppor that at this time.	The PatMan Careflow system is built using SNOMED terminology. When the user issues and exception to the workflow they are supposed to indicate the exception utilizing a SNOMED browser. When utilizing GUIDE the user does not interact with a controlled vocabulary	The AsbruView documents make no reference of support for or access to terminology services. No utitlities for loading or maintaining external terminologies are described.

Modeling & Encoding Process			
M. Mode of Operation – What is the general process to encode guidelines? Does it support multi-layered modeling that allows clinical experts to interact easily with knowledge experts?	The GLIF3 methodology is to have clinicians encode a top-level cnceptual view of a guideline and have knowledge engineer encode the computable parts. A third layer involves mapping and cutomization of encoded guidelines to deployment institutions. The GLIF workb bench does not yet support these layers.	The initial guideline model is created in GUIDE utilizing an algorithm tool to indicate the flow of logic. This is then converted to a Petri Net represention with another tool called Income (I'm checking it out). This model can then be entered into a simulator for analysis. Once the simulation is approved it can be ported to the Oracle production system. The knowledge engineer would interact with the PetriNet representation.	AsbruView is intended to be used by "non-technical" clinicians (after some initial training). Its metaphor-based UI allows clinicians to manipulate plan representations, and to interact with knowledge engineers during that process. AsbruView is designed to visualize the "clinical flow" for clinicians.
N. Multi-user support – What kind of multi-user support does it provide? Does a client software allow multiple remote users to work collaboratively?	No multiuser support.	Oracle workflow is an enterprise sized software tool that can support a large number of simultaneous users and jobs (patients). There is not any multiuser support or versioning built into GUIDE that I could find.	No multi-user support
O. Extensibility – How extensible is the system? Does it have a library of components that can be assembled in different ways? Does the database or programming environment create any known restraints of scale?	The GLIF workbench does not appear to be extensible.	GUIDE is a browser based tool built in Java. Based on this other modifications to the browser environment should be supported. The Run time environment can be extended with additional tables, Also the Workflow engine supports a separate programming language (?). This allows extensions to the decision making ability of the system to be written.	AsbruView appears extensible only with coding by its creators.

I. User-friendliness – How does it make it easy for domain experts to enter guideline knowledge? How well does it hide the complexities of the underlying guideline model? What visual metaphors does it use to aid the knowledge entry process? Are the component modes of operation understandable, scalable and useful for: **a. the clinical domain expert** **b. the knowledge engineer** **c. the software maintenance vendor?**	The GLIF workbench has an easy-to-use flowchart tool that automatically lays out a flowchart. The tool provides both flowchart view and a tree view of the components of a guideline. The flowchart view has two panes. Selecting an object in a flowchart in the left-hand-side pane automatically displays the attributes of that object in the right-hand-side pane. The form that displays attributes and their values allows hiding of attribute values.	PatMan utilizes two different representations of the guideline both an algorithm and a petri net. I don't know if this is a two way representation. In other words I don't know if changes to the petri net would be reflected back to the algorithm. The GL model is converted to the workflow representation without the intervention of the content expert. The knowledge engineer has access to the workflow representation built into Oracle Workflow Builder. The system them runs on the Oracle rule engine. Administrators can monitor the status of individual patient workflows.	No first hand experience available to us. However, K & M report a study in which 6 naïve physicians (after a 45-min training session) are asked to "author a plan for their every day work". They report that the physicians did "surprisingly well", and were able to understand the visual methaphors employed and to successfully manipulate plans.
P. Evidence – When entering guideline rules, is there a way to specify the references to medical literature and/or enterprise standards of care that justify the rules?	Yes. A user can associate "supplemental materials" with each guideline step	The guideline representation in GUIDE does not seem to have a slot for reference. The Oracle rule engine could be programmed to access references.	No.
Q. Does the software support maintenance of multiple versions with rollback and compare functionality?	No	Oracle is an industrial strength database engine. The tools built by … do not have versioning or roll back built in.	No.
Verification, Simulation & Localization			
R. Verification – What are the mechanisms to verify the guideline knowledge base? Internal scenario data integrity and consistency? Compliance with external vocabulary standards? Compliance with syntax standards for logic expression?	The tool supplies no verification mechanism.	GUIDE translates the guideline representaion into Petri Nets. The PatMan system then utilizes a Petri Net modeling tool to run simulations to verify the completeness of the system. Use of SNOMED maintains the vocabulary.	No verification or integrity checking available.

S. Simulation – Does it provide support for guideline simulation so that new guideline knowledge can be rapidly tested?	No	Yes, Guide outputs the guideline represenation as a Petri_net utilizing WPDL code. The developer then maps the tasks to the organizational ontology. A program called "Income is used to visualize the details fo the organizational unit. C30	No simulation functions available.
T. Localization – What kind of support does it provide for localizing a generic version of an encoded-guideline for particular institutions?	No	The guideline is represented in relational tables. This supports extension and local modification of the representation. It is unknown if there are tools to support localizaton directly.	AsbruView is well-suited for modifying (localizing) the visual representation of a plan. What is not clear is whether or not the underlying Asbru code is modified at the same time.
SUMMARY			
U. What are its strengths?	* The flowchart tool is easy to use. It automatically lays out the graph. • Having a flowchart view and tree view of a guideline is useful in understanding the structure of an encoded guideline. • Selecting a node in the flowchart automatically shows attributes of the node in adjacent pane facilitate browsing.	Incorporates workflow modeling into the guideline system therefore it can use a commercial workflow system (Oracle) for the knowledge engine. The front end is represented entirely in Java so it can run in any browser. Terminolgy is based on SNOMED and is enforce by requiring exceptions to the guideline to be represented in SNOMED.	• AsbruView was designed to provide for visualization of guideline plans to non-technical physicians. One value of AsbruView is that its developers specifically explored visual methaphors (e.g. the running track methaphor) that are familiar to non-technical users, and a departure from the "usual" way (e.g. flow charts, diagrams) of representing guideline structure. AsbruView may not be an ideal solution, but it stimulates us to think hard about visual alternatives for display of guideline content, and the authors direct us to Tufte's classic works on visual representation of information for ideas. • Another novel approach to AsbruView was the concurrent presentation of two views (Topological View and Temporal View) of the same guideline plan, allowing clinicians to see the general flow of the guideline simultaneously alongside the more detailed temporal representation. My impression is that while the Topological (running track) view was fairly intuitive, the Temporal view (Gantt chart like symbols) was not. • AsbruView appears to have a fairly robust ability to represent tempor • AsbruView has the ability to display a variety of relations between pla • AsbruView has the ability to represent and display the following plan

V. What are its weaknesses?	* The DSG GLIF workbench is very much a work in progress. Much of the GLIF3 guideline model are not yet supported.	Assumes that medical work process is represented through clinical practice guidelines and that an ontological description of the organization exists. Petri-Nets are complex models to understand. It isn't clear how well the algorithmic representation in guide translates into a production system. I suspect there is a great deal of custom knowledge engineering behind the scenes.	• AsbruView is apparently in use only at the original development site and has not had wide evaluation or use. My guess is that this tool would only be modifiable or extensible by the original devleopers. • AsbruView is described as a representation tool for "skeletal plans" - it was not clear from the papers if AsbruView could support encoding of the level of detail required for execution of guidelines in real CIS environment. Along these lines AsbruView does not appear to support a patient information model, nor does it have support for representing resources required for execution of a guideline. • I did not see any discussion of an ability of AsbruView to provide access to controlled terminologies. • Did not see evidence of capabilities to perform guideline simulation or integrity checking during authoring or encoding.
W. References	None	[1]Quaglini S, Stefanelli M, Lanzola G, et al. Flexible guideline-based patient careflow systems. Artif Intell Medicine 22 (2001) 65-80: [2]Dazzi L, Fassino C, Saracco R, Quaglini S, Stefanelli M. A Patient Workflow Management System Built on Guidelines. JAMA suppliment 2001. [3]Quaglini S, Fassino C, Stefanelli M, et al, Guidelines-based careflow systems. Artif Intell Med 20. (2000) 5-22 [4]Quaglini S, Dazzi L, Gatti L, Stefanelli M, Fassino C, Tondini C. Supporting tools for guideline development and disseminiation. Artif Intell Med. 14 (1998) 119-37. The developers web site.	This review is based on: (1) Kosara, R. & Miksch, S. Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans, and (2) Kosara, R. Metaphors of Movement -- A User Interface for Manipulating Time-Oriented,Skeletal Plans (Masters Thesis), 1999

	CG-AM	GEM Cutter Version	URUZ
	(Clinical Guidelines Acquisition Manager)		
General Information			
A. Purpose	**CG-AM** (**Clinical Guidelines Acquisition Manager**) is one of four "modules" in a comprehensive suite of guideline authoring, management, representation, and execution tools. The other three modules are: CG-KRM (Clinical Guidelines **Knowledge Representation Manager**); CG-EM (Clinical Guidelines **Execution Manager**); and CG-IM (Clinical Guidelines **Interface Manager**). CG-AM is designed to support original guideline authoring as well as encoding of already documented guidelines.	The Guideline Elements Model (GEM) is intended to serve as a document model for representations of the attributes of clinical practice guidelines (CPG) in a standard format. GEM Cutter is a tool for marking up existing text based guidelines the the GEM XML based ontology.	To gradually convert a large mass of clinical guidelines to semantically meaningful representations, we have developed a hybrid, multifaceted representation, an accompanying distributed architecture, the Digital Electronic Guideline Library, (DEGEL) and set of web-based software tools, which gravitates a set of guidelines gracefully from text-based, through structured text (segmented and labeled by Asbru semantic tags), to fully formal, machine- readable, executable representations
B. Target Users	CG-AM is designed to provide "expert" physicians with a user-friendly graphic interface to acquire guidelines into the CG-Knowledge Represenation Manager.	The GEM framework "is intended to be useful to developers, diseminators, implementers, amaintainers, and end users of guidelines."[2]	Developers who build guideline models. Knowledge engineers and domain specialists who enter guideline knowledge.
C. Institution / people – Who are the developers of the workbench?	Laboratoria di Informatica Clinica University del Piemonte Orientale Amedeo Avogadro Alessandria, Italy (Paolo Terenziana, Gianpaolo Molino, Mauro Torchio)	GEM is an outgrowth of a system in use at Yale. GEM Cutter was designed by the Yale Guidelines Review Group to support "logical analysis"-- the process by which "recommendation componenets are extracted from the natural language of a published CPG. Yale Center for Medical Informatics, Yale Guidelines Review Group. (Richard Shiffman, Abha Agrawal, Kristi-Anne Polvani, Bryant Karras, Aniruddha Deshpande, Peter Gershkovich)	Medical Informatics Research Center Department of Information Systems Engineering Ben Gurion University, Beer Sheva, Israel 84105 (Yuval Shahar M.D., Ph.D., research students at Ben Gurion University, Stanford University, and the Veterans Affairs Palo Alto Heath Care System, who assisted in assessing the tools.)
D. Time frame – When did the project start?	Guessing late 1990's.	The initial framework for the markup process began in 1995. Gem Cutter was developed to support markup. GEM appears to be first published in 2000 with GEM-Cutter being released about the same time.	This work has been done over the past 10 years with the original work done at Standford and subsequent papers first published in 1996.

E. Status – Is the project completed, ongoing…? Is the software a demo, a research prototype, commercial …?	In the 2001 paper, the authors indicate: (1) they have implemented CG-AM and CG-KRM prototypes using Java and Oracle, with partial implementation of the full feature set, and have used the prototype to model guidelines in several clinical domains. (2) They speculate about the possibility of combining efforts in the future to use GEM as an underlying guideline model, using XML representation. Current status of the project unknown.	"The GEM Document Type Definition (DTD) was balloted as an international standard for the representation of practice guidelines in XML format and will become ASTM standard E2210-02."[1] GEM Cutter is freely available.	The project is ongoing. URUZ is currently in beta testing with formal evaluation studies currently being conducted.
F. Availability – For those outside the project, are the workbench software and models freely downloadable, available under license, unavailable…?	Unknown	GEM Cutter version 1.3.1 is available for download at http://ycmi.med.yale.edu/GEM/ I was unable to find any licensing information but the software is copywritten 2000-2001	The project is not freely downloadable because it is in beta testing. Currently the user can potentially retrieve any guideline and edit it at will. There is no authoring control or auditing process in place.
G. Applications – How and where is the workbench being tested or used?	Prototypes of CG-AM and CG-KRM have been used to encode guidelines for: bladder cancer, reflux esophagitis, and heart failure. This prototype evaluation was conducted by the CG-AM developers, using physicians who had had some training on the tool.	GEM is being used for guideline appraisal and development at the American Academy of Pediatrics.	The workbench is being tested at Stanford under the direction Dr. Mary Kay Goldstein at the Palo Alta VA medical center. All testing will be done with patient de-identified data.
H. Installed base and numbers of users currently employing the software; the purpose of their use.	I am guessing that use is only by the original developers.	Unknown- will need to ask	The application is available over the web with approximately 15 users.
Components			

I. Guideline model – What is the underlying guideline model? Is the guideline model geared towards any specific types of guidelines?	The underlying guideline model is a "representation formalism" that underlies the CG-KRM (Clinical Guidelines Knowledge Resource Manager) module. Their stated goals for this representation formalism are that: (1) it be capable of representing guidelines across many different clinical domains, and (2) it allows expert physicians to represent all relevant clinical guideline knowledge in an understandable manner. The model includes actions (work actions, query actions, decision actions, conclusion actions), structural relations (e.g., is-a, has-part), and control relations (sequence, concurrency, alternative, repetition). In addition, it has a strong ability to model temporal relations specific to guidelines.	GEM Cutter is a tool for rendering text based guidelines in the Guideline Element Model (GEM). GEM is an XML DTD that contains all the attributes needed in a published guideline.	There is a specific underlying model present. The model seems to be based on an underlying classification system of guidelines. It is not geared toward any specific type of guideline.
J. What are the capabilities supporting, or supporting development of, the following guideline features:			
a. Enterprise workflow context and modeling	The CG "representation formalism" does not employ a specific workflow model. However, it can represent work actions (along with attributes of work actions), as well as the temporal and sequential relations among work actions. It can represent (as text) some description of the clinical context for a guideline. The model can also represent resource and/or cost limitations associated with a guideline.	GEM includes an attribute for the care setting but does not appear to contain other workflow specific information.	There is one node in the tree called process but this is generic and doesn't seem to have the ability to specify different workflow issues that would be required of a fully implementable guideline.
b. Information processing context and modeling	The CG "representation formalism" was designed with the aim of representing "contextual limitiations", such as availability of clinical and other resources.	By creating an XML document, the output of GEM Cutter should allow repurposing. However, there isn't any processing model implied.	None
c. Graphical (flowchart logic) depiction	The CG-AM tool provides a graphical view of guidelines. The Structure Window shows relations between guideline actions in a format similar to a Windows directory "tree" in the left panel, and a flowchart view (fairly similar to the graphical display in Protege) in the right panel.	GEM Cutter includes a flow chart layout of the attributes in the XML DTD. However the flow chart is not interactive with the actual XML. The logic of the guideline itself is not displayed graphically. It is possible to create a flow chart module that reads the input from the Algorithm attribute and displays them.	No graphical depiction of the workflow. Work is ongoing in this area and it is under development.

d. Data layer instantiation of logical elements into standard data elements	The CG-KRM can receive data from four databases: (1) the **Clinical DB**, which provides standard terminology for actions and conditions; (2) the **Pharmacological DB**, which provides a "structured list" of drugs and their costs (sounds like a formulary-GM); (3) the **Resources DB**, which lists resources (e.g. CT, NMR) available in a given hospital; and (4) the **ICD DB**, which contains the international coding of diseases. The CG-AM interacts with these databases to enforce use of standard vocabularies during the authoring/encoding process.	There is no connection between the GEM Cutter output and any actual instantiation. GEM does use standard nomenclature where available.	The data is instantiated into its own logical structure. There is no standard for representation of data elements that I am aware of. A standard is under development for representation of data elements.
e. Execution engine for run-time support?	The **CG-EM (Clinical Guidelines Execution Manager)** executes guidelines previously encoded by the CG-AM module. The CG-EM retrieves patient data at the time of execution; manages (e.g., start, stop, suspend) the execution of guidelines for individual patients; and records traces of "completed guideline executions" in the patient's clinical history. The during guideline management and execution CG-EM interacts with physicians via the **CG-IM (Clinical Guidelines Interface Manager)**, which is a user-friendly interface.	None	No, URUZ is a editing tool used in a suite of tools for guideline development and implementation. The purpose of URUZ is to allow a domain expert to convert a free text clinical practice guideline into a marked-up guideline through cut and past like gem cutter. The next function is to allow a knowledge engineer to further mark-up the text using ASBRU such that it could be implemented into a CIS for decision support or retrospective review for quality assessment.
K. EMR – What is the model of patient information?	There is a patient data model (**Patient DB)**. According to the authors, "the schema of the Patient DB parallels that of the Clinical DB [which] makes it possible to automatically retrieve from the Patient DB at execution time. Access to patient-specific data is obtained through "Query Actions" -- for data from: (a) patient history, (b) physical examination, or (c) laboratory results.	GEM Cutter is a markup tool that does not persuppose any EMR	There is no inherent model for patient information that I can discern.
L. Controlled Terminology Services – Does it provide access to controlled terminology services? How smooth is it to use standard terminologies when entering guideline knowledge? Are there utilities for loading and maintaining versions of external terminologies?	The CG-KRM can receive data from four databases: (1) the **Clinical DB**, which provides standard terminology for actions and conditions; (2) the **Pharmacological DB**, which provides a "structured list" of drugs and their costs (sounds like a formulary-GM); (3) the **Resources DB**, which lists resources (e.g. CT, NMR) available in a given hospital; and (4) the **ICD DB**, which contains the international coding of diseases. The CG-AM interacts with these databases to enforce use of standard vocabularies during the authoring/encoding process.	Since the actual contents of the guideline are not specified by GEM Cutter or the underlying GEM document model. The developer of the guideline and the user of GEM Cutter must enforce teminology control externally.	The help section discusses the use of LOINC etc. but I can not find any access to controlled terminology services within the application. There is a program to allow mapping to LOINC and as other standards are adopted. More follow-up with Uval is needed to better answer this question.

Modeling & Encoding Process			
M. Mode of Operation – What is the general process to encode guidelines? Does it support multi-layered modeling that allows clinical experts to interact easily with knowledge experts?	Physicians (with some training) use the CG-AM module to build guidelines as structures comprising actions. New actions are selected from a toolbar of action type icons presented by the graphical UI. Sub-windows pop up to allow users to enter the detail attributes of actions as well as the details of relations among actions. CG-AM supports "browsing" the details of guideline components already acquired, and also supports integration of controlled vocabularies during the encoding process, as well as internal consistency checking. CG-AM appears to support "multi-layered modeling" that would facilitate interaction between clinicians and knowledge engineers.	The user loads a text representation of the guideline (ASCII or RTF) into the left panel of GEM Cutter. They then highlight sections of text and apply attributes to that section. This is then either displayed in outline form with the attached attributes or can be displayed as Raw XML. There is no communication necessary between the content expert (the guideline) and the user of the system. It would be possible for the content expert to use the system without an intervening published guideline.	Multilayered modeling is strongly supported with its hybrid approach. There is ample opportunity for interaction between the knowledge engineer and domain expert.
N. Multi-user support – What kind of multi-user support does it provide? Does a client software allow multiple remote users to work collaboratively?	No evidence of multi-user support.	The output of GEM Cutter can be used by another author but there is no support for versioning or multiple users.	No, there is no source control editing environment that allows for multiple authors to work on a CPG simultaneously and follow changes that one author has made. Multiple user can access the application and work on it at the same time but it must be on different guidelines. More clarification from Uval is required.
O. Extensibility – How extensible is the system? Does it have a library of components that can be assembled in different ways? Does the database or programming environment create any known restraints of scale?	CG-AM appears to be extensible -- but with programming by the original developers. CG-AM is designed to be independent from other modules (e.g., CG-KRM, CG-EM, CG-IM). I could not ascertain if CG-AM itself was assembled from "sub-module" components. No information available on scalability.	There is no known intrinsic limit on the size of the guideline marked up. The author continues to add copies of attributes to encompass the entire guideline. GEM Cutter has no built in extensibility, no extra widgets, or authoring tools to add on.	URUZ is not extensible but more clarification is needed.

I. User-friendliness – How does it make it easy for domain experts to enter guideline knowledge? How well does it hide the complexities of the underlying guideline model? What visual metaphors does it use to aid the knowledge entry process? Are the component modes of operation understandable, scalable and useful for: **a. the clinical domain expert** **b. the knowledge engineer** **c. the software maintenance vendor?**	Impressions based only on reading the one reference paper: (1) Physicians (after brief training) were able to author a small number of new guidelines and encoded a small number of previously documented guidelines. The authors report that their guideline model was expressive enough to cover a variety of clinical algorithms.	For it's purpose GEM Cutter is easy to use with an adequate help file and a graphical navigation display. Unfortunately, the graphical display is not interactive with the guideline outline. The guideline must be in text format for importation into GEM Cutter. GEM Cutter does not appear to support embedding images in the outline. GEM Cutter uses an outline metaphor to display the GEM attributes with their attached text. GEM cutter is designed for use by the domain expert or knowledge engineer. Users mus be versed in the GEM DTD and make judgements about where text fits within the outline.	It seems very easy for domain experts to enter knowledge and it hides the complexity of the model very well. The URUZ tool seems to take a free text guideline and allows a domain expert to either create a guideline or cut and paste a guideline into a semi-structured format. The next step of converting this structure into an asbru marked up computable guideline and the model is not clear to me.
P. Evidence – When entering guideline rules, is there a way to specify the references to medical literature and/or enterprise standards of care that justify the rules?	No evidence of ability to represent references, etc.	Yes, one of the main purposes of GEM is to support all the information about the guideline in an organized structure. The actual logic of the guideline is a small part of the GEM Ontology. Every guideline step has associated data about the source, strength of evidence and other explanatory information.	No, it is not obvious to me.
Q. Does the software support maintenance of multiple versions with rollback and compare functionality?	Unknown.	No. However the XML files can be saved with different names and compaired with other XML utilities.	No.
Verification, Simulation & Localization			
R. Verification – What are the mechanisms to verify the guideline knowledge base? Internal scenario data integrity and consistency? Compliance with external vocabulary standards? Compliance with syntax standards for logic expression?	During authoring/encoding, CG-AM provides three types of consistency checking: (1) Name and range checking against values in the Clinical DB; (2) Logical consistency checking (e.g., are decision actions always preceded by query actions); and (3) Temporal consistency checking -- a semantic check of temporal constraints within the guideline (e.g. can overall duration specified contain all necessary actions).	GEM cutter and GEM do not enforce any standard on the contents of the attributes. Any free text entry is acceptable. A nonsense guideline can be easily represented in GEM	

S. Simulation – Does it provide support for guideline simulation so that new guideline knowledge can be rapidly tested?	No specific mention of this in the paper.	No	No.
T. Localization – What kind of support does it provide for localizing a generic version of an encoded-guideline for particular institutions?	The CG "representation formalism" was designed with the aim of representing "contextual limitations", such as availability of clinical and other resources. In addition it integrates a "Resource DB", which lists resources (e.g. CT, NMR) available in a given hospital.	No there is no customization except at the file name level.	There is no obvious support for localization.
SUMMARY			
U. What are its strengths?	• It is important to note that CG-AM is one of four modules in a comprehensive approach to guideline authoring, encoding, representation, and execution. While it appears that implementation of this approach to date is only partial, this R&D group has identified, and attempted to address many of the challenges that our SAGE project faces, including interaction between a guideline and patient-specific data. • It is important to note that objectives of the "CG" guideline model include representing guidelines across many different clinical domains, and representing not only work actions, but associated (complex) temporal and sequence relations as well. The researchers developed a thoughtful ontology and structure for guideline representation. • Also important for SAGE is that the "CG" guideline model was specifically designed to be able to represent "contextual limitations" -- clinical and other resources required to operationalize a guideline. • The CG-AM visualization views appear to resemble a combination of "Windows file tree" - display of actions, combined with a graphical display somewhat similar to that • CG-AM does have facilities for integrating controlled vocab	GEM is a balloted standard that represents all the attributes of a guideline needed for most administrative purposes. GEM Cutter is easy to utilize once the guideline is represented in text format. The GEM ontology is well thought out.	The strength of the system is related to its reliance on classification as a foundation for guideline creation and editing.

V. What are its weaknesses?	• The "CG" guideline environment appears to have had only limited implementation and use to date - and only at the developing institution. • The CG-AM authoring/encoding module's visualization interfaces are functional, but not necessarily easy to use (like Protege). • It does not appear that a goal of the "CG" project is interoperability across heterogeneous clinical information systems.	The GEM ontology is fixed within GEM Cutter. The user can add copies of an attribute but cannot add new attributes to the ontology. There is no versioning, localization, or multiuser capability inherent in the application. GEM Cutter only allows copying text pieces from the original to the GEM representation, but does not retain the connection to the source position. The GEM attribute diagram is not interactive and does not serve as a navigation tool in what can become large source files. Due to the need to represent the guidelines in text format much formatting is lost as well as diagrams.	The underlying flaw with this tool to me seems to be that it is dependent on the free text guideline for encoding of CPG. I do not think it is safe to assume that a free text guideline can be encoded and practically implemented into a CIS system. The classification system does contribute to problems with implementation though. It allows polyheirarchy of CPG which may cause problems with system implementers knowing when and where a particular guideline is to be used and not used.
W. References	Terenziani P, Molino G, Torchio M. A Modular Approach for Represneting and Executing Clinical Guidelines. Artificial Intelligence in Medicine 23 (2001), 249-276.	[1] Karras BT, Nath SD, Shiffman RN. A Preliminary Evaluation of Guideline Content Mark-up Using GEM--An XML Guideline Elements Model. Proc AMIA symp; 2000, 413-417 [2] Shiffman RN, Karras BT, Agrawal A, Chen R, Marenco L, Nath S. GEM: A Proposal for a More COmprehensive Guidleine Document Model Using XML. JAMIA 7(5) 2000, 488-498	Shahar Y. A Hybrid Framework for Representation and Use of Clinical Guidelines. Proc. of the AMIA Annual Symposium, 2002; (submitted). http://medinfo.ise.bgu.ac.il/DeGel/