A knowledge representation architecture for the construction of stories based on interpretation and evidence

Susan W. van den Braak and **Gerard A.W. Vreeswijk** [1]

Abstract. This paper describes *Stevie*, a knowledge representation architecture for the analysis of complex legal cases. *Stevie* is targeted at legal professionals who may use it to infer stories (plausible and consistent reconstructions of courses of events) from evidence and hypotheses. *Stevie* is based on known argument ontologies and argumentation logics.

1 INTRODUCTION

This paper describes *Stevie*, a knowledge representation architecture for making sense of evidence through stories and their justification. This system is targeted at criminal investigators who may use it to gain a better overview of complex cases. In the process of making sense of large quantities of data, it will enable crime investigators to formulate their hypotheses as stories of what might have happened and to make their underlying reasoning explicit.

In project meetings with crime investigators we learned that in the analysis of crime cases there is a demand for a support tool that offers the ability to search and combine large quantities of data. In fact, crime investigators already use powerful search tools to match possibly relevant data. What they seem to lack is functionality with which search results can be interpreted, explained, and related to each other in a larger context. *Stevie* is a first stab at the realization of such facilities.

With respect to argument visualization, the contribution of *Stevie* is threefold. Firstly, it represents cases (among others) as di-graphs rather than trees. Thus, unnecessary duplication of nodes is avoided. Further, *Stevie* possesses an inferential component to incorporate predefined argumentation schemes. This component also assesses the dialectical status of nodes to suggest plausible stories to analysts. Finally, it represents temporal information and is thus able to rule out stories that are temporally inconsistent.

2 SYSTEM PURPOSE

This section describes the context in which *Stevie* operates. It also describes the functionality that the system provides at its interfaces.

2.1 Context

Stevie provides support during criminal investigations by allowing case analysts to visualize evidence and their interpretation of that evidence in order to construct coherent stories. It allows them to maintain overview over all information collected during an investigation, so that different scenarios can be compared. Moreover, they are able to express the reasons why certain evidence supports the scenarios. In this way it may help them in seeing patterns, discovering inconsistencies and identifying missing evidence.

It must be emphasized that *Stevie* is not meant to be used in the preparation of trials; nor is it intended as a tool for modelling legal cases, since police and prosecution have different responsibilities. Crime analysts are supposed to follow promising leads, without too much concern about proving guilt in court. Once one or more suspects are determined, the prosecution takes over and *Stevie* drops out of the picture.

2.2 System interface

Stevie is presented as a web front-end to an SQL database (Fig. 1). Users log in and create a case record, or select a case which they want to work on. Each case is presented in a split screen where the upper half displays a global overview of the case and the lower half displays the attributes of a node that is selected by the user in the upper half of the screen.

The case can be visually represented through multiple views. These views include a graphical view, a table view, a hierarchical view, a report view, a summary view, and a linear view. The report view is a verbal and linear dump of the case representation and can be used as an official print-out for off-line instantiations (think of the need for paper files and communication by traditional mail). *Stevie* draws heavily on ideas from visualizing argumentation [6, 11]. Therefore, the graphical view is considered to be most representative for on-line uses of *Stevie*.

If a node is clicked in the upper half of the screen, its contents (and some of its other attributes) can be edited in the lower half of the screen. Nodes can be created in isolation (bottom-up) or hierarchically through other nodes (top-down). Thus, a case is built.

2.3 State of implementation

Stevie is prototyped in *Aafje*. *Aafje* is programmed in PHP and stores case data in a PostgreSQL database. *Aafje* has the following functionality: creation of cases, support of multiple users, linkage to quotes in PDF documents, usage of schemes, creation of nodes top-down (from the main claim), bottom-up (from evidence), and by scheme instantiation. Unimplemented features include a properly working labeling algorithm for stories and a faithful incorporation of the AIF ontology (to be explained below).

[1] Department of Information and Computing Sciences, Utrecht University, the Netherlands

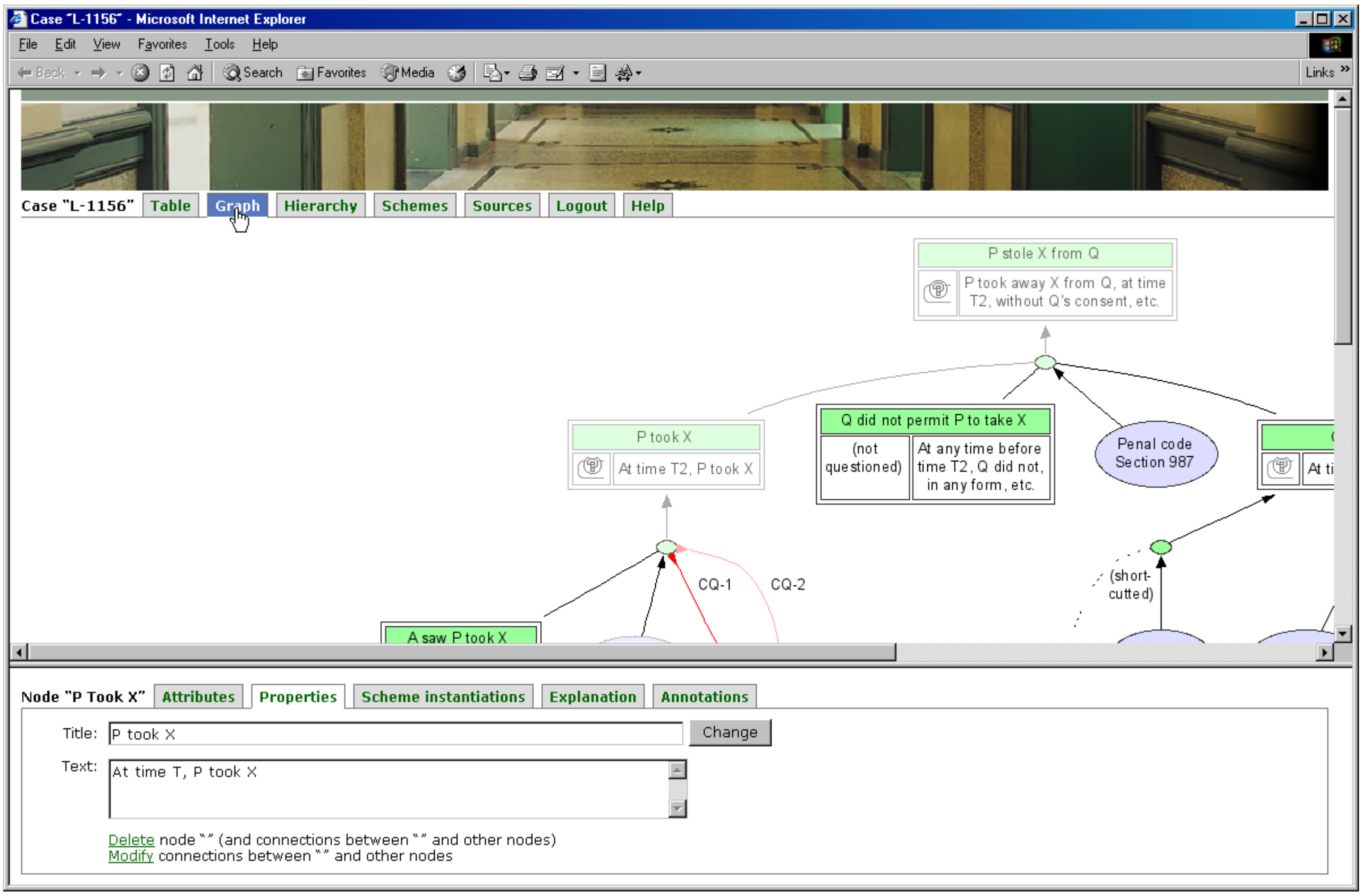

Figure 1. Screenshot of system interface.

3 THEORY

Stevie's conceptual framework is to a large extent based on the core ontology for argument entities and relations between argument entities as described in a recent document on argument interchange formats (AIFs) and ontologies [18]. According to the AIF ontology, knowledge about a (not necessarily legal) case is stored in two kinds of nodes, viz. *information nodes* (I-nodes) and *scheme instantiation nodes* (S-nodes). I-nodes relate to content and represent claims that depend on the domain of discourse. In Fig. 2, I-nodes are rectangles (and conversely).

	Green	*Red*	*Blue*
Rectangle (I-node)	P-node	P-node	Q-node
Ellipse	S-node	S-node	Scheme node

Figure 3. Node visualization.

Schemes

According to the AIF standard, I-nodes may be connected to indicate inferential support, and S-nodes represent justifications for those connections. S-nodes (small red or green ellipses in Fig. 2) are instantiations of general inference schemes (large blue ellipses) and are called scheme instantiation nodes (or instantiation nodes for short). Table 3 summarizes node visualization. Schemes are pre-defined patterns of reasoning, such as rules of inference in deductive logics but then broadened to non-deductive logics or domain dependant patterns such as witness testimony in evidential reasoning [11, 3, 12].

In principle schemes are predefined and may be reused by case analysts. There are many schemes and our system cannot contain them all. Currently, *Stevie* uses the scheme list of Araucaria [10] which to our knowledge is the first system that deals with schemes.

Stories

According to Wagenaar *et al.*'s theory of anchored narratives [16], a story is a credible, coherent, temporally consistent, and defensible set of claims that together describe a possible course of events of a case that is subject to investigation.

Stevie uses defeasible reasoning [4, 9] to distill stories out of large quantities of information. If we use principles of defeasible reasoning to define stories, we may say that stories must be contained in conflict-free and self-defending collection of claims (I-nodes). A set of claims is conflict-free if (and only) if it does not contain a conflicting pair of I-nodes. The meaning of conflict-freeness is further defined in the subsection on stories (p. 4). A set of claims is self-defending if (and only if) every argument (made up of I-nodes and S-nodes) against an element of that story can be countered with an argument made up of I-nodes that belong to that story. In addition to defeasible reasoning principles we add a third constraint on stories, namely that they must be temporally consistent. What this means is defined below. A simple example of a case representation that contains valid stories is shown in Fig. 4.

4 STRUCTURE

The most important elements of *Stevie* are nodes and links between nodes.

Nodes

The basic building block of *Stevie* is a node. A node is an elementary piece of information that is used in modeling cases. Nodes can be facts in a case or claims about a case and are typically displayed in a GUI. Every node possesses two mandatory attributes, viz. a title field and a text field. Additionally nodes possess optional (scalar) attributes such as slots indicating time and location, the name of the analyst who created the node, and a list of records of all edits. Finally, a node can refer to zero or more real-world objects, such as persons, institutions, locations and cars.

I-nodes fall apart into two categories, namely, quotation nodes (Q-nodes, colored blue) and interpretation nodes (P-nodes, colored green and red, depending on the party of interest, cf. Fig. 2).

Quotation nodes

A quotation node represents information from outside the system, such as quotes from testimonies, reports, minutes and other original source documents, but also plain data such as car registration details, addresses, and telephone numbers. The text field of a quotation node is a literal transcription of the selected fragment and cannot be further edited. Once imported, the content of a quotation node is fixed, and its status is incontestable within the system.

There are two types of quotation nodes: information nodes and scheme quotation nodes (scheme nodes, for short). Information quotation nodes (blue rectangles) are ordinary quotations from external source documents. Scheme nodes (blue ellipses) represent a special type of external information, namely, (quoted) argumentation schemes.

Interpretation nodes

A P-node represents an observation or claim made by a user for the purpose of making sense out of quoted data. Nodes that (indirectly) support the main thesis are colored green; nodes that (indirectly) contest the main thesis are colored red, and nodes that may serve both interests are colored yellow. In the present example, yellow nodes do not occur but they may occur in more complicated cases.

Interpretation nodes can be questioned by users and can be supported by other nodes. Unquestioned interpretation nodes provide support of themselves. Questioned interpretation nodes (indicated by the blue question mark on the left) need further support from other nodes in order to be "believed" or "IN" (the evaluation of nodes is described below). Whether this support indeed exists depends on further input of case analysts.

Thus, an I-node may contain a quote from a source document (Q-node), or it may contain an explanation or interpretation of such a quote (P-node).

Schemes

Schemes belong to a special group of nodes that represent predefined patterns of reasoning. A single scheme describes an inference, the necessary prerequisites for that inference, and possible critical questions that might undercut the inference. A scheme may be instantiated to one or more scheme instances (S-nodes). Graphically, an S-node is depicted as a small ellipse that is red or green depending on the side of interest. Every S-node springs from a scheme node (blue ellipse) and uses zero or more antecedent nodes to justify a consequent node (cf. Fig. 2).

As an example of how schemes may be applied, consider Fig. 2. If a case analyst wishes to support the claim that "P stole X from Q", *Stevie* will present one or more inference schemes from which this conclusion follows. In this case, the analyst chose the scheme entitled "Penal code Section 987". According to this scheme, in order to prove "P stole X from Q", it is necessary to prove three sub-claims, viz. "Q owns X", "Q did not permit P to take X", and "P took X". In this case, these three claims suffice to conclude that "P stole X from Q".

Schemes can also be instantiated the other way around, from quotation (or interpretation) nodes to conclusion nodes. Consider again Fig. 2. If an analyst wants to find out which conclusion follows from the testimonial evidence "A: "I saw P took X"", he may chose the "Quote instantiation" scheme and will be automatically presented with the conclusion that follows being "A said: "I saw P took X"".

Most schemes incorporate a pre-defined list of so-called *critical questions*. A critical question is a possible circumstance that may invalidate a particular scheme instantiation [11, 12]. Thus, critical questions are latent rebutters of S-nodes or, put differently, latent undercutters. Fig. 2 shows examples of critical questions for some schemes. For instance, the inference from "A saw P took X"" to "P took X" through "Perception" may be rebutted by the knowledge that A is short-sighted and did not wear glasses.

Links

To create a network of inferential and temporal interdependencies, nodes can be linked through two types of connections, that is, inferential connections (arrows and arrows with reversed arrowheads in Fig. 4) and temporal connections (arrows with solid dots as arrowheads).

Inferential links

Inferential connections can be created by instantiating schemes. Thus, although inference links and S-nodes look different, they are actually the same. Supporting connections are displayed by arrows, attacking connections by reversed arrowheads.

Temporal links

Temporal connections are made when two nodes possess sufficient information to relate them temporally, or else when a case analyst decides that two nodes must be connected temporally. Once temporal connections exist it is possible to represent stories of what might have happened as a sequence of temporally structured nodes.

Two nodes receive a temporal connection automatically if they both possess an explicit time stamp. Nodes can be connected manually as well. If a case analyst decides that node A precedes node B in time, he creates a temporal link between A and B. In doing so, the case analyst must qualify that link by indicating his own confidence in that link. This qualification can be selected from a predefined set of modalities (for example: "certainly," "beyond a reasonable doubt," and "likely").

Stories

The objective of *Stevie* is to create, on the basis of quotes and interpretations, possible stories that indicate what might have happened. In *Stevie*, a story is a set S of nodes that satisfies the following two postulates:

1. S is conflict-free and self-defending.
2. The underlying temporal digraph T of S is internally consistent (i.e., acyclic) and consistent with temporal and causal orderings implied by scheme instantiation nodes.

Thus, S must be conflict-free, self-defending, and temporally consistent. Since all information available in a case together is almost always inconsistent, it is usually the case that a single case yields room for multiple stories. Based on inferential connections, nodes can be evaluated as being "IN" or "OUT". Quotation nodes and unquestioned interpretation nodes are "IN".

There exist several semantics for node evaluation. *Stevie* uses the grounded and the admissibility semantics, respectively [5, 9]. For the sake of simplicity, only the admissibility semantics is briefly quoted here [5]. This semantics enforces the two properties that are mentioned under (1) above.

Nodes can be either "IN," "OUT," or "UNDEC" (undecided).

1. A questioned interpretation node N is "IN", if it satisfies the following two conditions.
   (a) N is supported by an S-node that is "IN"
   (b) All S-nodes that attack N are "OUT"
2. A questioned interpretation node N is "OUT", if it satisfies one of the two following conditions.
   (a) All S-nodes that support N are "OUT"
   (b) N is attacked by an S-node that is "IN"
3. A questioned interpretation node N is "UNDEC", otherwise.

More complex configurations possess more than one valid labeling, and in some configurations the empty story (all nodes "UNDEC") is also a valid labeling. When instantiating a scheme, newly created antecedent elements cannot have been questioned yet so that they are "IN", until either the corresponding S-node or else one of its antecedent nodes is either questioned or attacked. In Fig. 2 the node "Q sold X to P" is out since it is undercutted by "P is a party concerned". As a result, the node "Q owns X" is "IN", because its rebutter is "OUT".

A detailed description of the algorithms used for graph "consistency checking" (as it is called by one of the reviewers) is beyond the scope of this paper. More detailed descriptions a various such algorithms can be found in the formal argumentation literature [4, 9].

5 RELATED WORK

As remarked in Sec. 2.2, *Stevie* draws heavily on ideas from visualizing argumentation. Compared to traditional issue-based information systems (IBISs) and argument visualization tools, however, *Stevie* is more directed towards the construction of stories than to visualization as a goal in itself. Further, *Stevie* uses a node ontology that is in line with the current standards on representation formats for argument interchange (AIF).

Because of its graphic interface, *Stevie* is strongly connected to FLINTS [7, 8, 19]. FLINTS (Forensic Led Intelligence System) is a methodology and software system that helps analysts to identify relevant information in large amounts of data. The difference between FLINTS and *Stevie* other than that FLINTS is a much more matured system, is that FLINTS is not centered around the construction of stories as *Stevie* is.

With respect to the data model, *Stevie* follows the same approach as case analysis tools such as Araucaria [11] and Legal Apprentice [17]. Araucaria is a software tool for the analysis and visualization of arguments. It supports argumentation schemes, and depicts arguments as trees of nodes, where nodes consists of quotes from a fixed text that is displayed in the left margin. Legal Apprentice (LA) is a case analysis system that visualizes evidence in so-called legal implication trees. Those are AND/OR tree-structures where nodes can receive a true, false or undefined status from case analysts. The main conceptual differences between *Stevie* and these systems is that *Stevie* uses a logic and ontology of which basic principles such as scheme instantiation [11, 3, 12] and admissibility [5] have a solid theoretical underpinning in the theory of formal argumentation [4, 9, 18].

With respect to argumentation and legal narratives, *Stevie* is also strongly connected to MarshalPlan [13], a formal tool to prepare legal cases for trial. The main difference between *Stevie* and MarshalPlan is that *Stevie* is more directed towards investigation than towards the preparation of legal trials.

Particularly relevant to mention is DAEDALUS [2], a tool that may help Italian magistrates and prosecutors in their work; it is not, like *Stevie* graphically oriented but its usefulness resides in the facility that it may be requested to validate and document steps made by the magistrate and the police.

A last approach that is interesting to mention is the coherentist approach as advocated by Thagard *et al.* such as ECHO [14, 15] and especially ConvinceMe [1]. The latter is an artificial pedagogical assistant to help students structure, restructure, and assess their knowledge about often controversial situations. Like *Stevie* it is a sense-making tool to formulate hypotheses based on evidence, but then based on principles of coherence rather than being based on principles of argument.

Acknowledgement. The authors like to thank Floris Bex, Henry Prakken and the anonymous reviewers for their constructive help. They also gratefully acknowledge the support from the Netherlands Organisation for Scientific Research (NWO) through project 634.000.429 and from the EU through project IST-FP6-002307 (ASPIC).

REFERENCES

[1] Stephen T. Adams, 'Investigation of the "Convince Me" computer environment as a tool for critical argumentation about public policy issues', *Journal of Interactive Learning Reseach*, **14**(3), 263–283, (2003).

[2] Carmelo Àsaro, Ephraim Nissan, and Antonio A. Martino, 'Daedalus: An integrated tool for the italian investigating magistrate and the prosecutor. a sample session: Investigating an extortion case', *Computing and Informatics*, **20**(6), 515–554, (2001).

[3] Floris J. Bex, Henry Prakken, Chris A. Reed, and Douglas N. Walton, 'Towards a formal account of reasoning about evidence: argumentation schemes and generalisations', *Artificial Intelligence and Law*, **11**, 125–165, (2003).

[4] Carlos I. Chesñevar, Ana G. Maguitman, and Ronald P. Loui, 'Logical models of argument', *ACM Comput. Surv.*, **32**(4), 337–383, (2000).

[5] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games', *Artificial Intelligence*, **77**(2), 321–357, (1995).

[6] *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*, eds., Paul A. Kirschner, Simon J. Buckingham Shum, and Chad S. Carr, Springer-Verlag, London, 2002.

[7] Richard M. Leary, *Evidential Reasoning and Analytical techniques in Criminal Pre-Trial Fact Investigation*, Ph.D. dissertation, University College, London, 2004.

[8] Richard M. Leary, John Zeleznikow, and Wim VanDenBerghe, 'User requirements for financial fraud modeling', in *Proceedings of BILETA 2003: British and Irish Law, Education and Technology Association 18th Annual Conference*, (2003).

[9] Henry Prakken and Gerard A.W. Vreeswijk, 'Logical systems for defeasible argumentation', in *Handbook of Philosophical Logic*, volume 4, 218–319, Kluwer, 2nd edn., (2002).

[10] Chris A. Reed and Glenn W.A. Rowe, 'Araucaria: Software for argument analysis, diagramming and representation', *Int. Journal of AI Tools*, **14**(3-4), 961–980, (2004).

[11] Chris A. Reed and Douglas N. Walton, 'Applications of argumentation schemes', in *Proceedings of the 4th OSSA Conference*, Ontario, Canada, (2001).

[12] Chris A. Reed and Douglas N. Walton, 'Towards a formal and implemented model of argumentation schemes in agent communication', *Autonomous Agents and Multi-Agent Systems*, **11**(2), 173–188, (2005).

[13] David Schum, 'Evidence marshaling for imaginative fact investigation', *Artificial Intelligence and Law*, **9**(2/3), 165–188, (2001).

[14] Paul Thagard, *Coherence in Thought and Action*, MIT Press, Cambridge, MA, 2000.

[15] Paul Thagard, 'Causal inference in legal decision making: Explanatory coherence vs. bayesian networks', *Applied Artificial Intelligence*, **18**(3/4), 231–249, (2004).

[16] Willem A. Wagenaar, Hans F.M. Crombag, and Peter J. van Koppen, *Anchored Narratives: Psychology of Proof in Criminal Law*, St Martin's Press / Prentice-Hall, 1993.

[17] Vern R. Walker. Early legal apprentice software (published as screencast). `http://people.hofstra.edu/faculty/vern_r_walker/LegalReasoning.html`, 2006.

[18] Steven Willmott, Gerard Vreeswijk, Matthew South, Carlos Chesñevar, Guillermo Simari, Jarred McGinis, and Iyad Rahwan, 'Towards an argument interchange format for multiagent systems', in *Proc. of the 3rd Int. Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006)*, (2006).

[19] John Zeleznikow, Giles Oatley, and Richard M. Leary, 'A methodology for constructing decision support systems for crime detection', in *Proc. of the Ninth Int. Conf. on Knowledge-Based and Intelligent Information and Engineering Systems (KES)*, pp. 823–829, (2005).

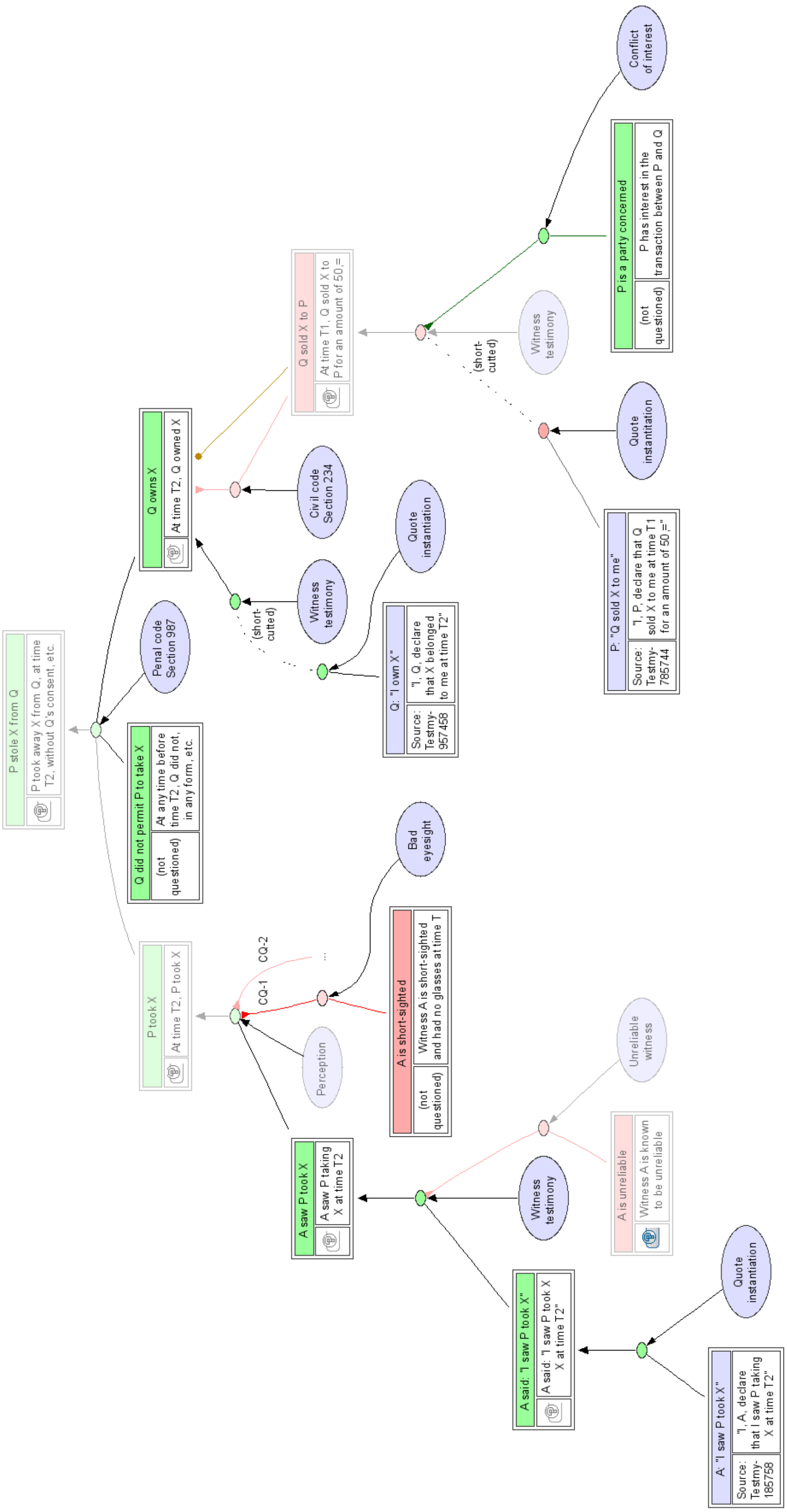

Figure 2. Graph view of theft case.

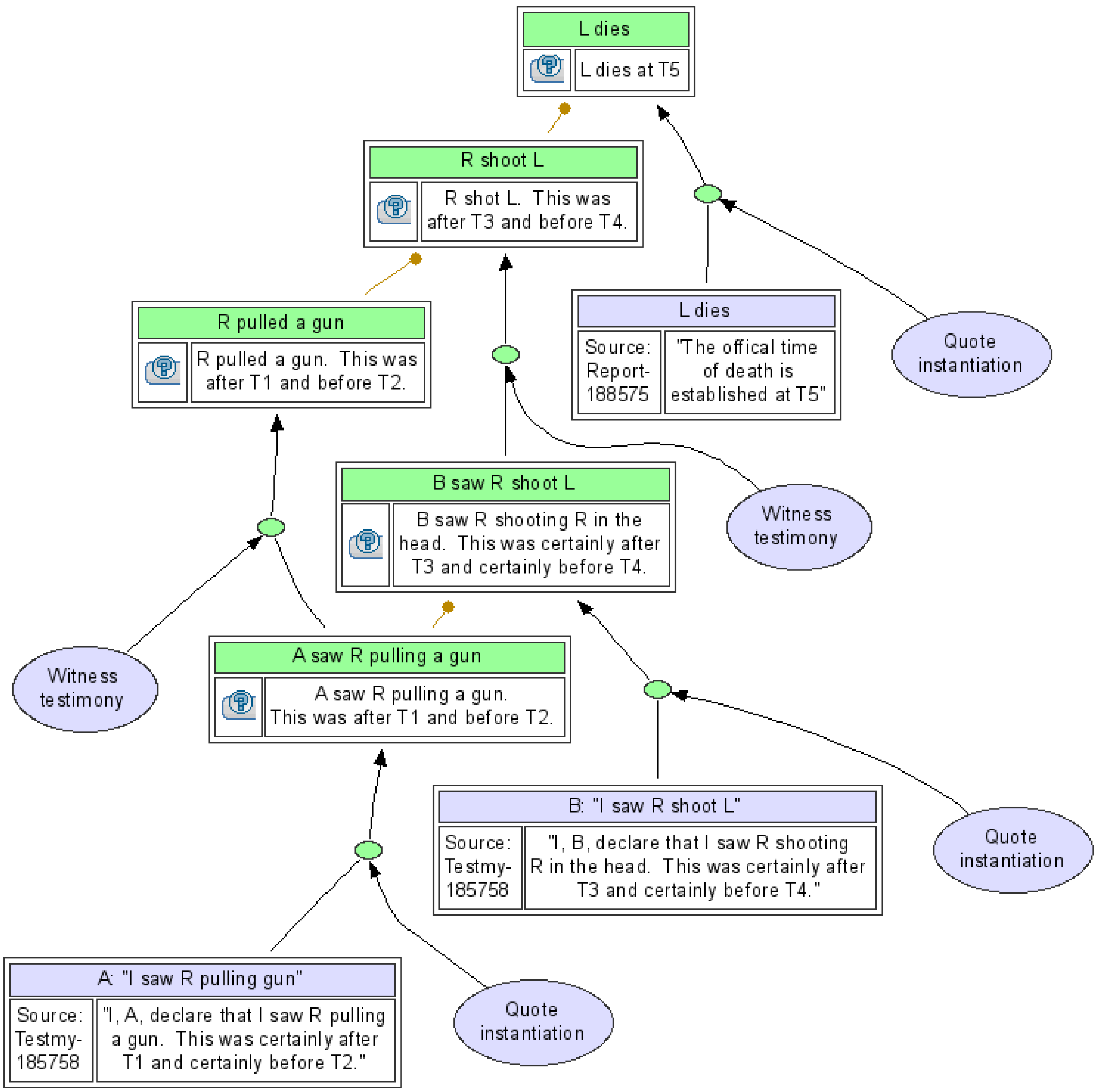

Figure 4. Graph view of shooting case.