

# **Seminar Design Rationale: Automated Argument Assistance**

Max Berger <max.berger@xs1t.de>

Juli 9th, 2001

## Overview

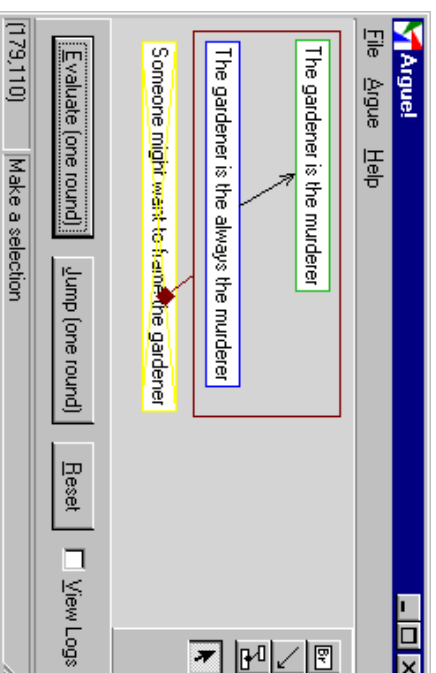
- Argument Assistance Tools
- Using ArguMed 2.0
- Usability of ArguMed for Software Design
- Restrictions when using ArguMed
- Bibliography

## **Argument Assitance Tools**

- **Argue!**
- **ArguMed**
- **ArguMed 2.0**

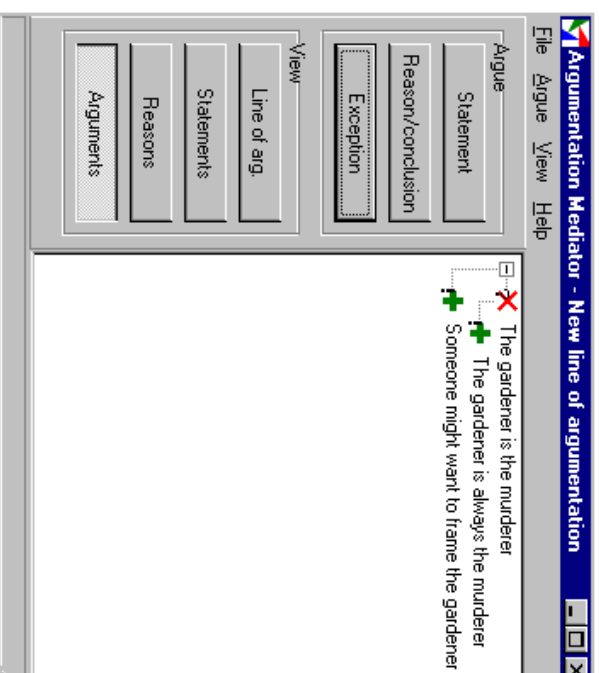
# Argue!

- First version of Bart Verheijs Argument Assistance tool
- Argument graphically shown
- Attack not
- Unfamiliar Design
- Written June 1998



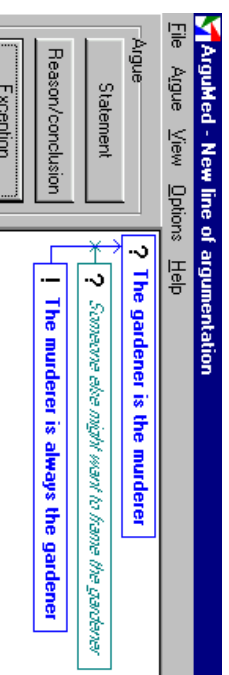
# ArguMed

- New underlying argumentation system
- Counterarguments are not directly shown
- Written September, 1998



## ArguMed 2.0

- Current Version
- Argumentation system slightly changed
- Integrated view
- September 1999



## Using ArguMed 2.0

- Introduction
- Type of Statements
- Justification of Statements
- Attacks
- Sample Session

## Introduction

- Based on Statements
- Statements can be linked
- Automatic justification



## Type of Statements

### Issues

- Marked with a ?
- Uncertain until justified

### Assumptions

- Marked with an !
- Already justified

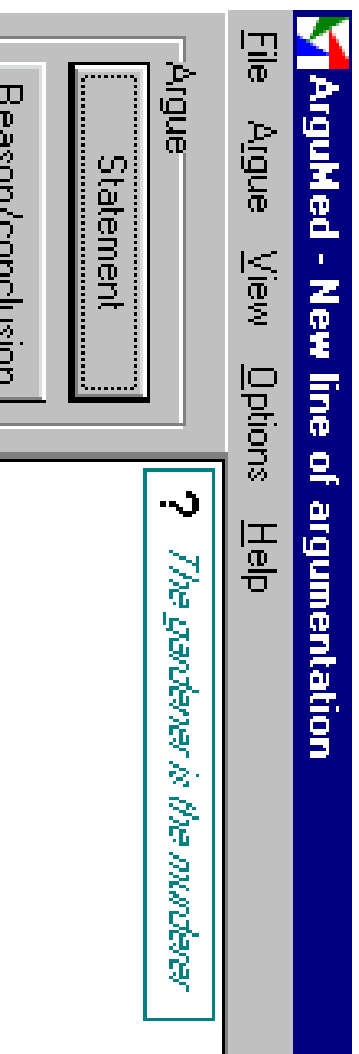
## **Justification of Statements**

- **Justified Statements are in bold font**
- *Unjustified Statements are in italic font*
- **Assumptions are always certain**
- **Issues are justified, when at least one statement leading to it is also justified**

## Attacks

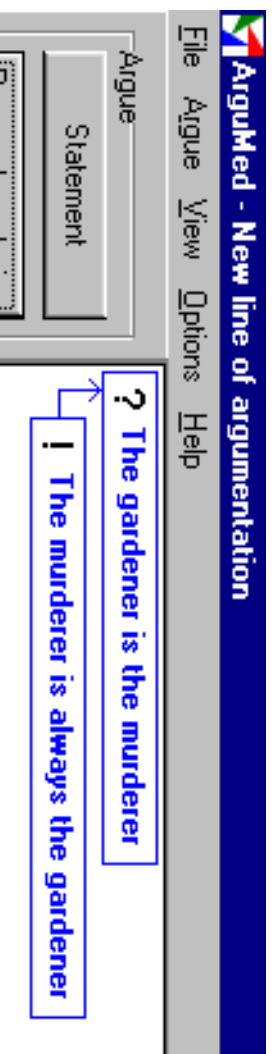
- Only Conclusions can be attacked
- An attack is successful when the attacking statement is justified
- A successful attack breaks this line of argumentation

## Sample Session



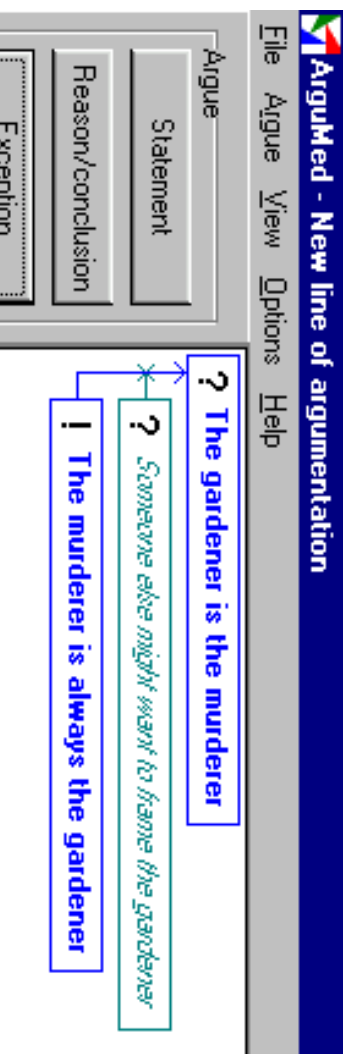
- We start with a single Issue Statement

## Sample Session (cont.)



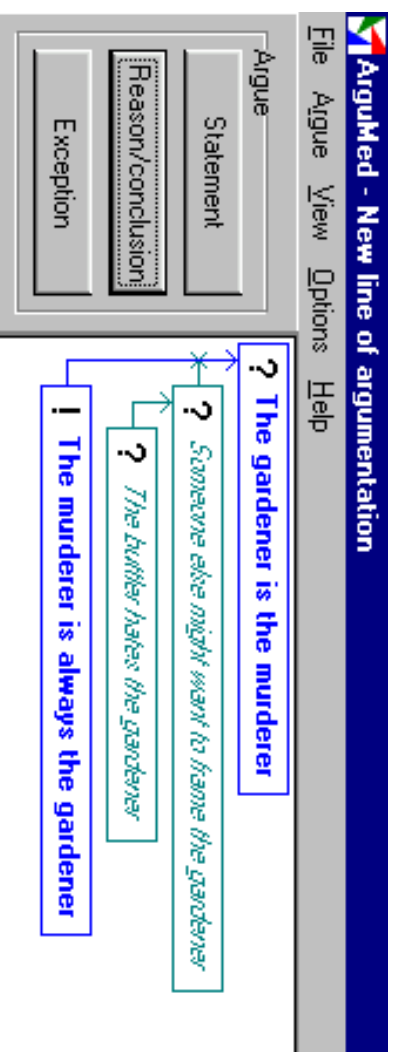
- Since this is just an Issue, it needs to be justified by an Assumption Statement.
- The original statement is now in bold, it is justified

## Sample Session (cont.)



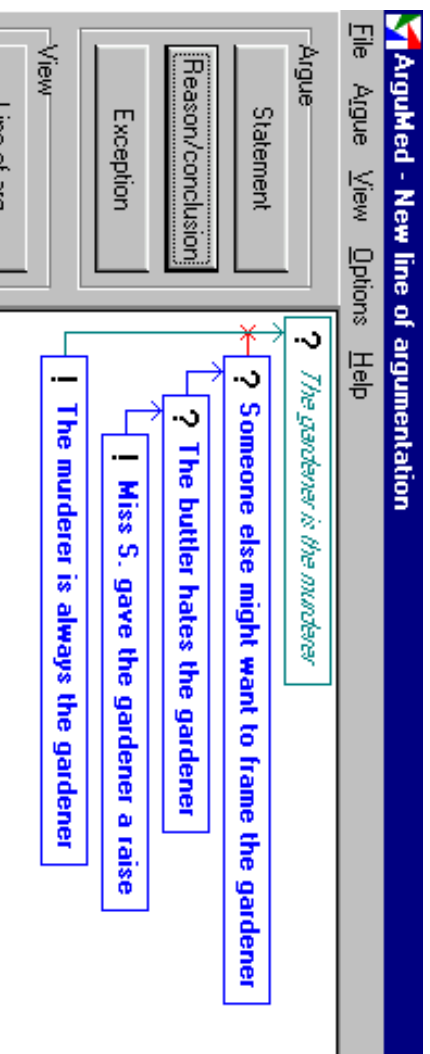
- Conclusions can be attacked by Exception Statements

## Sample Session (cont.)



- The Attack is only successful when the Exception itself is justified.
- Issues can only be justified by Assumptions

## Sample Session (cont.)



- Now the attack is justified
- The original statement is now no longer justified



## Sample Session (cont.)

The screenshot displays the ArguMed software interface. At the top, a menu bar includes 'File', 'Argue', 'View', 'Options', and 'Help'. Below the menu is a toolbar with buttons for 'Argue', 'Statement', 'Reason/conclusion', and 'Exception'. A 'View' section contains buttons for 'Line of arg.', 'Statements', and 'Arguments'. The main workspace shows a hierarchical argument structure. The root node is a question: '? The gardener is the murderer'. A child node is another question: '? Someone else might want to frame the gardener'. This node has three children: '! The butler had his day off', '? The butler hates the gardener', and '! Miss S. gave the gardener a raise'. The 'butler hates' node has two children: '! The cook was out at the disco' and '! Miss S. gave the gardener a raise'. The 'cook was out' node has one child: '? The cook hates the gardener'. The 'cook hates' node has one child: '! Miss S. gave the gardener a raise'. The 'gardener has an alibi' node has one child: '! The murderer is always the gardener'. Red 'X' marks are placed above the root node, the 'butler had his day off' node, the 'gardener has an alibi' node, and the 'The murderer is always the gardener' node. A green box highlights the 'Someone else might want to frame the gardener' node and its children.

## **Usability of ArguMed for Software Design**

- Original purpose of ArguMed
- How ArguMed supports Design Decisions
- ArguMed vs. QOC
- Meeting Scheduler Example

## Original purpose of ArguMed

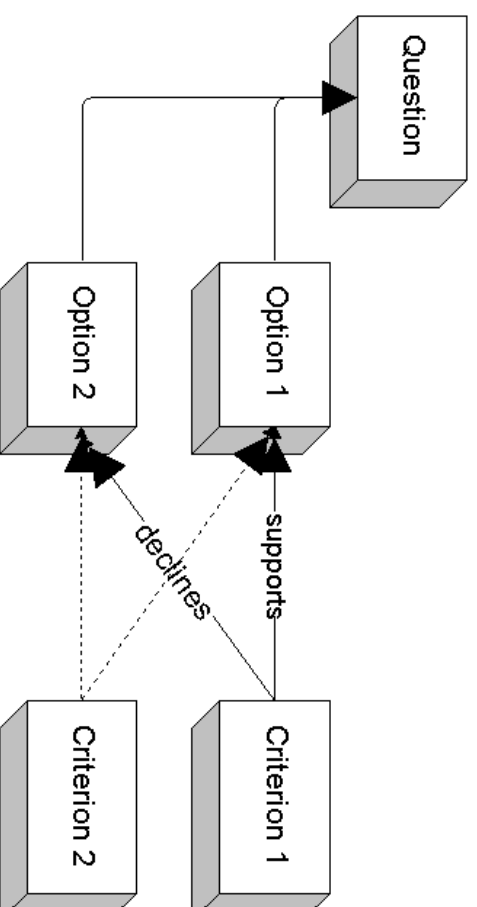
- Structure arguments
- Show dependencies between different arguments
- Show line of argumentation
- For use by lawyers

## **How ArguMed supports Design Decisions**

- Lines of argumentation can be clearly shown
- ArguMed is very simple to use
- Automatic justification helps a lot

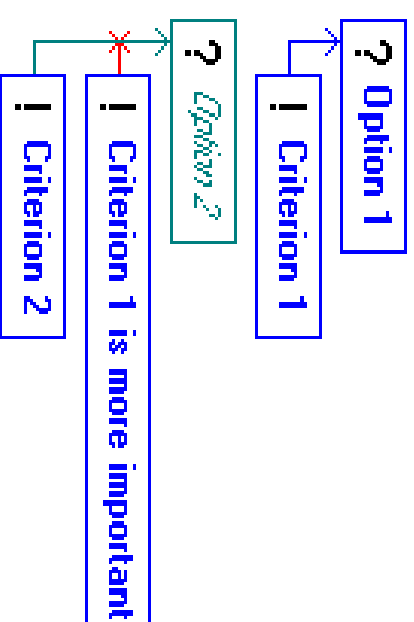
## ArguMed vs. QOC

- The QOC Model:



## ArguMed vs. QOC (cont.)

- The Question has no equivalent in ArguMed
- Options correspond to Issues
- Criteria correspond to Assumptions



QOC model with ArguMed

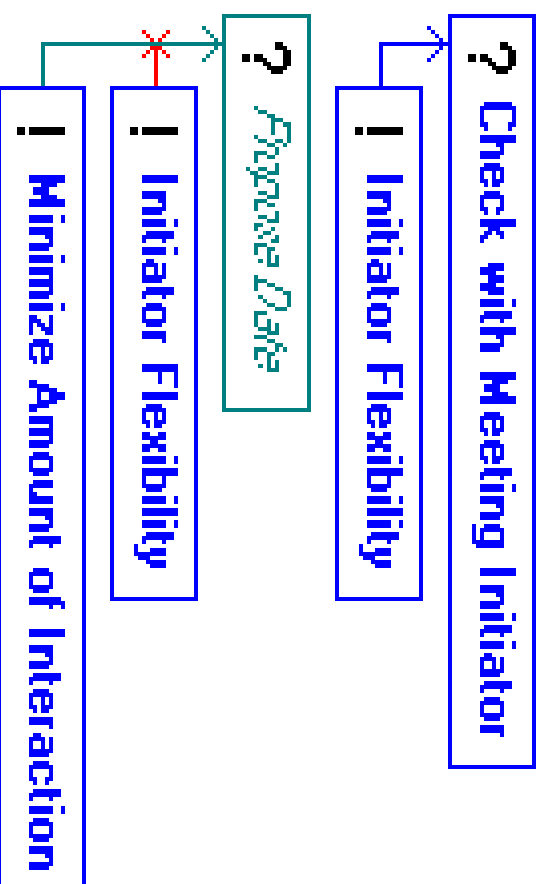
## Meeting Scheduler Example

Justification: Use Case Input participant information

Options	Criteria	
	<u>Minimize Amount of Interaction</u>	<u>Initiator Flexibility</u>
<b>The system doesn't automatically propose a date but first checks with the Meeting Initiator.</b> ( <a href="#">thiede, 6/29/01 7:30 AM</a> )	-	+
The system proposes a date as soon as possible (i.e. as soon as data is available from every participant) ( <a href="#">thiede, 6/29/01 7:31 AM</a> )	+	-

## Meeting Scheduler Example (cont.)

Same with ArguMed:





## **Restrictions when using ArguMed**

- Missing Features in ArguMed
- ArguMeds interface

## Missing Features in ArguMed

- Only Conclusions can be attacked, not statements itself
- Its not possible to pick one of several options and still show rationale for the other
- There could be a weighting system (points) for arguments
- ArguMed is build for cases where there is one real solution

## ArguMeds interface

- Goal was a modern interface
- Statements can not be edited afterwards
- Attacks / Conclusions have to be chosen from a dropdown list, cannot be clicked

## Links / Bibliography

- <http://www.metajur.unimaas.nl/~bart/aaa/>
- Automated Argument Assistance for Lawyers, Bart Verheij, presented at ICAIL 99  
<http://www.metajur.unimaas.nl/~bart/papers/pdf/icail99.pdf>