VisClim - Visualisation of Climatological Data

Student Project

Markus Kobel

January 2004

Supervised by:

Dr. Michele Lanza
Prof. Dr. Oscar Nierstrasz

Institute of Computer Science and Applied Mathematics
University of Berne, Switzerland

The address of the customer:

Climatology/Meteorology Research Group
University of Bern
Institute of Geography
Hallerstrasse 12
CH-3012 Bern

The address of the author:

Markus Kobel
Hindelbankstrasse 21
CH-3325 Hettiswil

or

Software Composition Group
University of Bern
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10
CH-3012 Bern
kobel@iam.unibe.ch
http://www.iam.unibe.ch/˜kobel/

Abstract

In this project, we developed a tool for generating videosequences from geographical data (temperature, sea level pressure, 500hPa geopot., ...). The customer for this application (called VisClim) is the Climatology/Meteorology Research Group (Climet) from the Institute of Geography of the University of Berne. They maintain a database which contains earth science data for the time span from 1500 until today. VisClim is able to load netCDF files and generate videosequences from the data in these files.

Contents

List of Figures

Chapter 1

Introduction

The application called VisClim (Visualisation of Climatological data) along with this document is the result of a project work of the Software Composition Group in conjunction with the Climatology/Meteorology Research Group of the University of Berne.

The purpose of this document is it to give the reader a short and simple overview of the technical aspects of the application and also to provide a user manual for VisClim.
In Chapter 2, the customer will be introduced and the purpose of VisClim and the used technologies will be illustrated. There are also some technical explanations about the used components and their role in VisClim. One can find a documentation for the user of VisClim in Chapter 3. All functions and preferences of VisClim are described there. It illustrates how to load files, generate pictures and videos, and how to handle all other features of this application. Finally, there are some conclusions of the development process and some personal impressions in Chapter 4.

Chapter 2

Description

2.1 Customer - Motivation for this Project

The Climatology/Meteorology Research Group (Climet) collected and restored a lot of geographical data for the time span from 1500 until today. All this data is stored in a database. People from the Climet group work with this informations and make calculations based on this data. They were not able to visualise these information in a simple and useful way. Until now, they just made some pictures of interesting points in time. The idea of this project is to visualise the information using pictures and movies.

2.2 Used Components

In this application there are three main components that work together: the Grid Analysis and Display System (*GrADS*), the Java Media Framework (*JMF*) and *VisClim* itself. Figure 2.1 shows how these components are connected. As input, VisClim can handle different kind of earth science data (in form of netCDF files). Then it generates other netCDF files out of the original ones and creates also script files which can be processed by GrADS. GrADS builds up the requested pictures according to the script files. After that, VisClim takes these pictures and generates *QuickTime movies* out of them using the JMF.

In the following sections, all of these components are described in more detail.

2.2.1 GrADS

We needed a tool for generating pictures from the geographical data. We have chosen the freely available *Grid Analysis and Display System (GrADS)*[1] because that application is well known by the people of the Climet group and they have been using it for a long time.

GrADS is an application which can visualise earth science data. It can handle different kinds of data, such as the netCDF format. GrADS may be run in batch mode (without graphics output window) and then process script files. So, we just generate the

[1] `http://grads.iges.org/grads/`

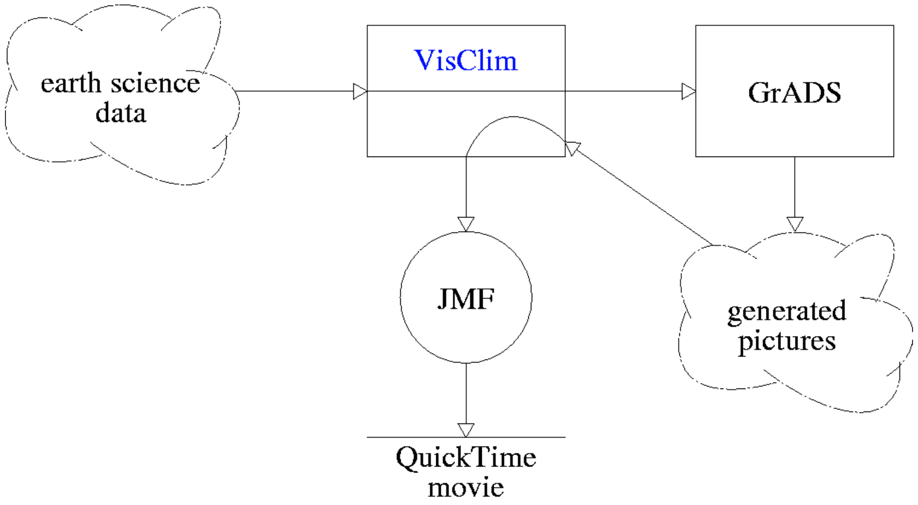


Figure 2.1: Overview of used Components.

needed script file with VisClim and then run GrADS in batch mode in the background. Unfortunately, GrADS is quite an old application without the possibility to access its functions through an API.

2.2.2 JMF - Java Media Framework

We also needed something that is able to generate video files. We have chosen the *Java Media Framework API (JMF)*[2] which is freely available.
The JMF is a package which can handle time-based media like video and audio. It provides various functionality to capture, playback, stream, and transcode multiple media formats. For this project we only used the possibility to generate a video file out of a bunch of picture files. [SUN 99]

2.2.3 NetCDF Data Format

We had to decide which data format we want to support primarily. We picked out the netCDF data format because in the future, this will be the most used format for earth science data as the people of the Climet group said. There is also a *NetCDF Java Library*[3] freely available which we used in VisClim as well.

NetCDF is intended for ”self-documenting data”. There are different ways to structure the data in a netCDF file. A netCDF file is a collection of dimensions, attributes, and variables:

- **Dimension:** A dimension is a named array index which consists of a name and a length. The length may be ’unlimited’ - if this is the case, the length can increase,

[2] http://java.sun.com/products/java-media/jmf/
[3] http://www.unidata.ucar.edu/packages/netcdf-java/

otherwise it is immutable. There may also be a 'coordinate variable' associated with a dimension - this is a variable with the same name as the dimension, whose single dimension is that dimension (for example: float lat(lat)).

- **Attribute:** An attribute is a (key, value) pair where the key is a String and the value is a String, Number or Number[] (in this context valid subclasses of Number are Byte, Double, Float, Integer or Short).
- **Variable:** A variable is a multidimensional array, which has a name and a collection of dimensions and attributes. Variables are the place where the data is stored. The data will only be read out of the file, when you request it (not already when you load the file). This circumstance makes it possible to read the data out of the file through a simple stream, for example through a http-stream from a remote server.

2.2.4 NetCDF Java Library

Through the Java API it is possible to read out all the needed data from a netCDF file and to write new files. The restriction is that once the structure of a file is written it is not possible to alter it. That means, after defining variables and its dimensions, it is only possible to fill up this structure with data, but it is for example not possible to define another dimension for a variable anymore. [CARO 03]

2.3 VisClim - Visualisation of Climatological Data

VisClim is supposed to become a helper application for the people of the Climet group. The goal is to generate pictures and movies from their earth science data. The graphical user interface of VisClim should be simple and the whole application should offer high usability.
VisClim may then reduce the time that is needed between getting earth science data and gaining an overview of the meaning of that data.
VisClim uses, executes, and sometimes depends on all of the components mentioned before. It would be possible to use VisClim without the JMF to generate pictures (but not videos). If GrADS is not available, VisClim can only be used to truncate netCDF files (without the possibility of visualising any data).
VisClim can be seen as the glue between the individual components. The user uses a simple interface to generate the pictures and videos without the need of writing script files or using some painful command line interfaces.

2.4 Technical Program Run

The purpose of this section is it to describe in a short way, how VisClim works and in which order the different components are used when we want to generate a video file.

1. When we load a netCDF file into VisClim, there will be instantiated a *NetCDF-Wrapper* which is realized as a singleton in this application, because there was no need to load more than one file at the same time.

2. After specifying the data to visualise and starting the video generation, VisClim reads the selected intervals of time, latitude and longitude and gives them to a new instance of a *NetCDFWriter* and a *ScriptWriter*.

3. These writers create a new netCDF file and a script file for GrADS. This new netCDF file only contains the data of the selected sections.

4. As soon as these files are available, GrADS will be executed in batch mode (no graphics output window is opened) with the generated files as input. When GrADS processes the script file, it generates the needed pictures.

5. VisClim waits until all the needed files are generated. Then, a list of these pictures is given to an instance of a *VideoWriter*, which will concatenate these pictures to a movie using the JMF. This video writer is executed in a new thread in order not to block the user interface of VisClim.

6. When the movie is finished, the writer notifies VisClim of this fact. Depending on the options that we have chosen in the generate video dialog (see also Section 3.6), VisClim will either open the movie in a new frame with its internal player or execute the preferred player with the movie file as an argument or VisClim will do nothing at all.

2.5 System Overview

We used the *Java 2 Software Development Kit (J2SDK), Standard Edition, Version 1.4.2_X*[4] in order to develop VisClim.
The whole system consists of 6280 lines of code. That code is divided into several packages. The most important ones (without test- and exception-packages) are shown in Table 2.1.

Package	**Total Classes**	**Abstract Classes**	**Concrete Classes**	**Afferent Couplings**	**Efferent Couplings**
visclim	1	0	1	0	5
visclim.controller	8	1	7	1	10
visclim.io	7	0	7	3	22
visclim.util	1	0	1	4	4
visclim.view	27	0	27	3	18
visclim.view.util	4	0	4	1	3

Table 2.1: Overview of the most important packages of VisClim.

Efferent Couplings: The number of other packages that the classes in the package depend upon is an indicator of the package's independence.
Afferent Couplings: The number of other packages that depend upon classes within the package is an indicator of the package's responsibility.

[4] http://java.sun.com/j2se/1.4.2/

2.6 Possible Extensions

There are many things that could be done in order to extend the usefulness of VisClim. Here are some ideas:

- Importer for different formats (gdx, lola, ...).
- Support for netCDF files which contain variables with more than three dimensions.
- Support for different video output formats (avi, mpeg, ...).
- Use of other visualisation tools than GrADS (FERRET, ...).
- Optimisation of the video generation (needed space, needed time).

Chapter 3

User Documentation

3.1 Main View

When VisClim has been started, the *MainView* (as shown in Figure 3.1) is displayed on the screen.

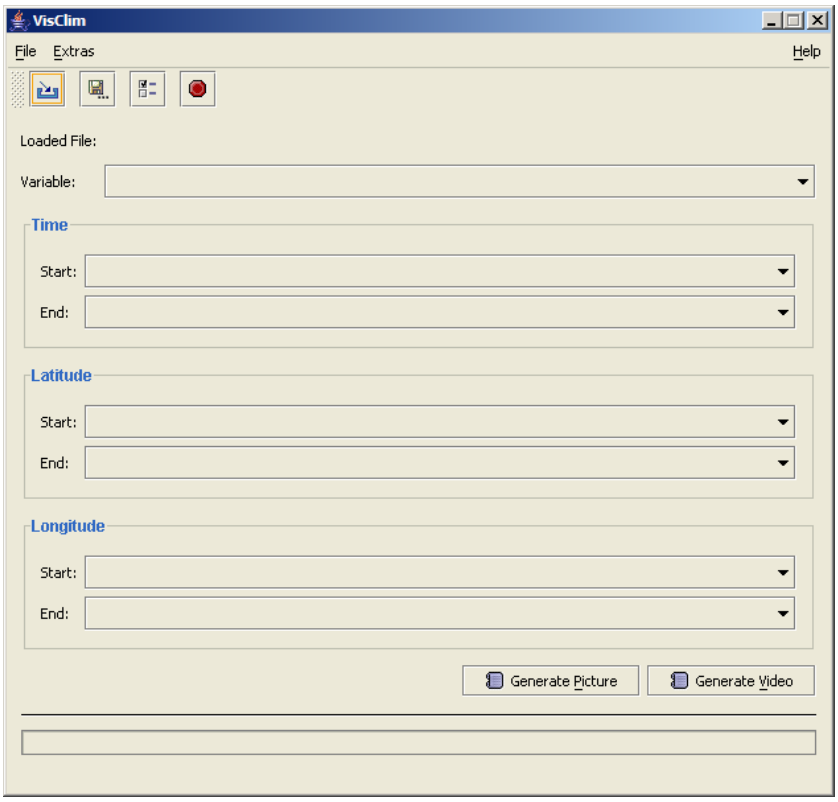


Figure 3.1: MainView of VisClim after launching.

Figure 3.2 shows that the MainView is separated into different parts:

1 The menu bar of this application.

2 The toolbar with some icons on it for fast access to the most used functions (load a netCDF file, save a netCDF file, open preferences window, exit VisClim). The toolbar may be dragged to another place if needed.

3 The path of the file which is actually loaded is displayed here.

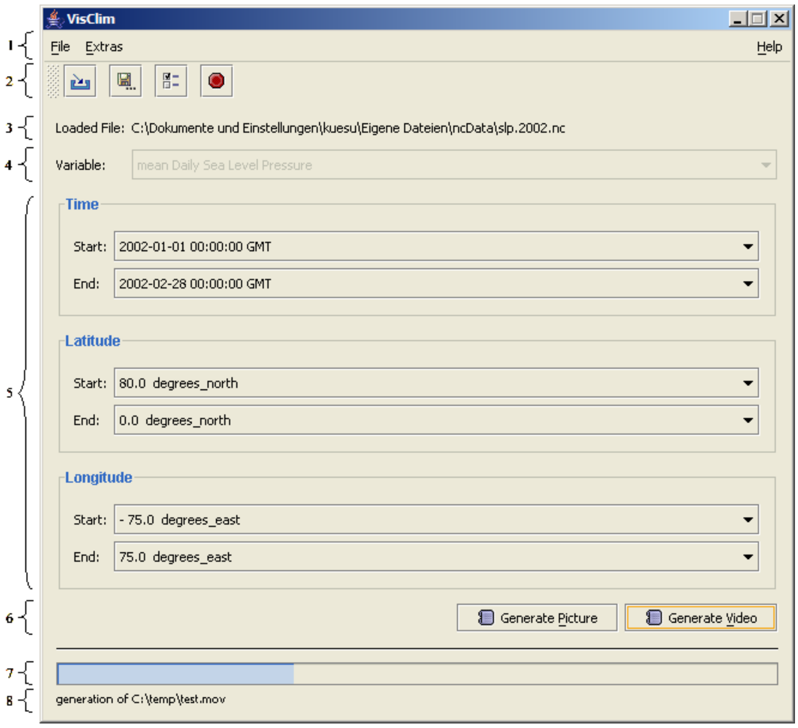
4 This combo box is only used, when the loaded netCDF file contains more than one variable that can be used for visualisations. If there is only one such variable, this combo box will be disabled by VisClim.

5 There are three similar parts: combo boxes for choosing the start and end value of *Time*, *Latitude* and *Longitude*. With them, we can choose the time interval to use for generating a video file and we have control over the region on earth which will be visulised.

6 The two buttons which are used for generating *pictures* and *videos*.

7 The progress bar shows the state while generating video files.

8 The status bar shows some information about the state of VisClim or about the actions that are performed.



Loaded File: C:\Dokumente und Einstellungen\kuesu\Eigene Dateien\ncData\slp.2002.nc

Variable: mean Daily Sea Level Pressure

Time

Start: 2002-01-01 00:00:00 GMT

End: 2002-02-28 00:00:00 GMT

Latitude

Start: 80.0 degrees_north

End: 0.0 degrees_north

Longitude

Start: - 75.0 degrees_east

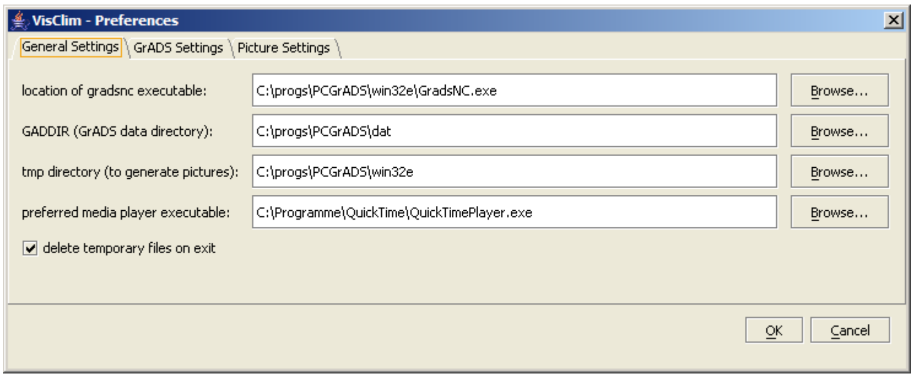
End: 75.0 degrees_east

generation of C:\temp\test.mov

Figure 3.2: MainView of VisClim while generating a video file.

3.2 Preferences

The Preferences of VisClim are partitioned into three sections: general settings, GrADS settings and picture settings.



location of gradsnc executable: C:\progs\PCGrADS\win32e\GradsNC.exe

GADDIR (GrADS data directory): C:\progs\PCGrADS\dat

tmp directory (to generate pictures): C:\progs\PCGrADS\win32e

preferred media player executable: C:\Programme\QuickTime\QuickTimePlayer.exe

☑ delete temporary files on exit

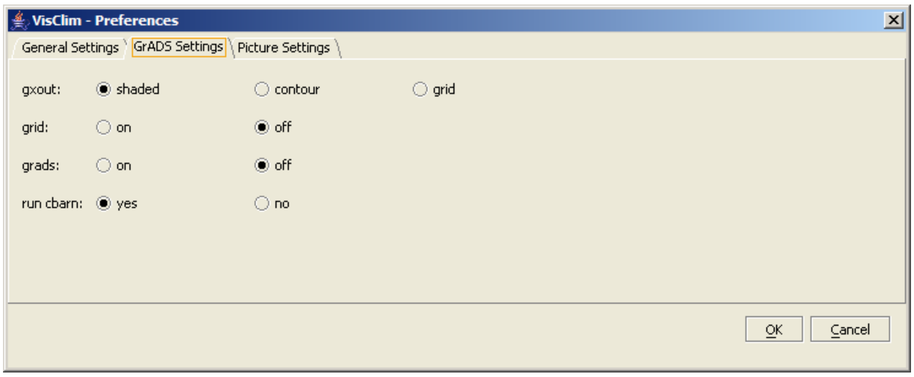
Figure 3.3: Preferences - General Settings.

As shown in Figure 3.3, there are the following options to set in the general settings tab:

- The location of the gradsnc executable: this is the binary of the GrADS package which can handle netCDF files. This is called by VisClim in order to generate the pictures.
- The GrADS data directory (GADDIR): this environement variable is needed by GrADS in order to find the world maps which we can see on every picture generated by GrADS. VisClim will set this variable when executing GrADS.
- The tmp directory: this is a directory where VisClim can store temporary files. There is a restriction on Windows platforms for this: there we have to choose the directory where the gradsnc executable is in. VisClim helps on this and chooses this directory when we change the location of the gradsnc executable (only on Windows).
- The preferred media player executable: you can choose to view the generated video files with this player. It must be possible to view QuickTime-movies with this player. Otherwise, it will fail to show the video files.
- Delete temporary files on exit: if we choose this option, VisClim will delete all temporary files when we shut down the application.

In Figure 3.4 we see all options of the GrADS settings tab:

- There are three possibilities for the *gxout* setting:
  1. *shaded:* The color will be drawn in areas over the pictures.
  2. *contour:* Only lines will be drawn.



gxout: ● shaded ○ contour ○ grid

grid: ○ on ● off

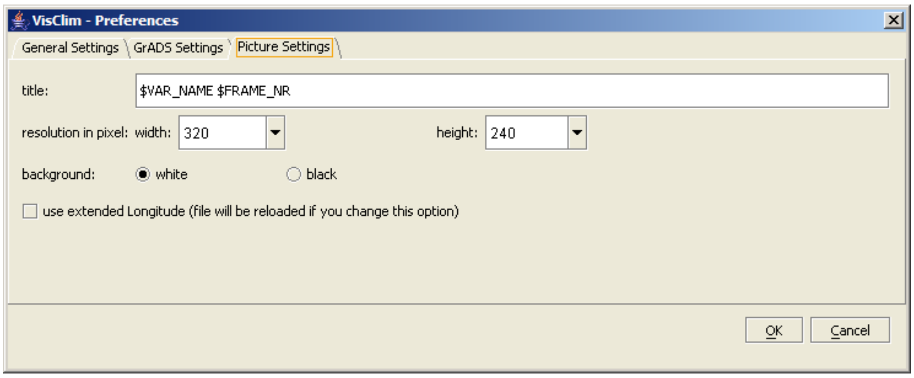
grads: ○ on ● off

run cbarn: ● yes ○ no

Figure 3.4: Preferences - GrADS Settings.

3. *grid:* There will be drawn squares with the values inside.

- The *grid* can be turned on or off.
- *Grads* can be turned on or off. If this is on, then GrADS will print its name and the date and time of the picture generation on every image.
- If *run cbarn* is set to yes, there will be a color bar printed on every image. This will only work when the gxout setting is set to shaded.



title: $VAR_NAME $FRAME_NR

resolution in pixel: width: 320 height: 240

background: ● white ○ black

□ use extended Longitude (file will be reloaded if you change this option)

Figure 3.5: Preferences - Picture Settings.

There are also some picture specific options to set, as shown in Figure 3.5:

- As *title* for the pictures we can choose any value plus some special variables. The variables will be replaced by VisClim. The following variables are possible:
  - $VAR_NAME will be replaced by the name of the actual variable.
  - $FRAME_NR will be replaced by the number of the frame in the video (if we just produce pictures, this will be 1 for every picture).

  - $DATE will be replaced by the right date/time value. The values will be the same as the ones we see in the MainView on the time axis.

- The *resolution in pixel* of the pictures. There are some standard resolutions to choose (320x240, 640x480, 800x600, 1024x768, 1280x1024). They may be changed to any value by the way.

- The *background* color of the pictures can be set to white or black.

- The option *use extended longitude* is used for duplicating the longitude values. With this it is possible to get pictures which show the whole world twice. The restriction here is, that the loaded file must contain longitude values which go from 0 to 360 degrees.

3.3 Loading Files

3.3.1 NetCDF

In order to load a netCDF file into VisClim, we can choose *Load NetCDF File...* from the *File* menu or click on the appropriate icon on the toolbar. The file we want to load has to be a netCDF file. If it is not such a file, VisClim won't open it.
We may load almost every netCDF file into VisClim, but we can't generate visualisations from all of them. If we want some visualisations, the file has to contain at least one variable with exactly three dimensions named *time*, *lat*, and *lon*.

3.3.2 Video

We can open some video files in VisClim itself and look at them with the internal player of this application. The internal player is very simple. In order to open a video file, we just choose *Open Video...* from the *File* menu. The video file will be opened in a new frame.

3.3.3 Picture

We may display all generated images in VisClim. If we choose *Open Picture...* from the *File* menu a filechooser will appear. The file we choose will be opened in a new frame.

3.4 Storing Files

3.4.1 NetCDF

Storing netCDF files is used for truncating large files. If we load a file into VisClim and choose *Save NetCDF File...* from the *File* menu (or click on the appropriate icon on the toolbar), VisClim will store just the selected sections (time, latitude, and longitude) of the selected variable into the new file.

3.4.2 Video

VisClim uses a standard path for storing its generated video files (they will be stored in the tmp directory). However, this path may be changed, before a video will be generated. Please refer to the Section 3.6.

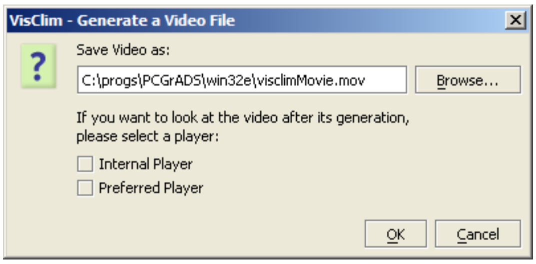
3.4.3 Picture

Generated Pictures may be stored as png files. In order to do that, we just choose *Save as...* from the *File* menu of the frame which contains the image we want to store. This makes it possible to generate some pictures and then (probably after comparing them) only store some of them.

3.5 Generating Pictures

For generating a picture, we only have to click on the *Generate Picture* button in the MainView (see also Figure 3.1) after loading a netCDF file into VisClim. VisClim will always take the *Start Time* value for knowing which time to use for the generated picture. It also takes into account the values of *Latitude* and *Longitude*.

3.6 Generating Videos


VisClim - Generate a Video File

Save Video as:

C:\progs\PCGrADS\win32e\visclimMovie.mov

If you want to look at the video after its generation,
please select a player:

□ Internal Player
□ Preferred Player

Figure 3.6: Generate Video Dialog.

When we click on the *Generate Video* button, VisClim takes the given time interval for generating a video file. Before the generation starts, a dialog will open up which is shown in Figure 3.6. Here we may choose another path for storing the file than the standard path. Also, we may choose to look at the video immediately after its generation. To do so, we just have to select a player for viewing the file. We have the choice between the *Internal Player* of VisClim and the *Preferred Player* which we can set in the preferences (see also Figure 3.3). If there is no preferred player set, this option won't be available.

3.7 Example of a Program Run

The purpose of this section is it to describe, what we have to do when we want to generate a video file.

1. We click the *Load a NetCDF File* icon and choose the file which contains the data we want to visualise.
2. We specify the time interval and the region (latitude and longitude) and click then the *Generate Video* button.
3. The dialog shown in Figure 3.6 is opened and we specify another location for the video file and choose to look at movie after its generation with our preferred player.
4. When the video file is finished, our preferred player opens up and shows the generated movie.

Chapter 4

Conclusions

4.1 Development Process

The development of VisClim started in the beginning of July 2003 and ended in January 2004. We used the *Concurrent Versions System (CVS)*[1] as our code repository. For building the system, we first used *Ant*[2] and as the application evolved we switched to *Maven*[3]. It was pleasant to work on the project with the knowledge that we can go back to any prior version if needed. With the help of Maven, we were also regularly informed about the actual state of the project, recent changes, compile problems, and many more (for more informations, we refer to the Maven homepage, where all plugins are described).

When the project started, we wanted to have a running prototype as soon as possible in order to see that all the components we had chosen (see also Chapter 2) work together. In order to reach that goal, we first created animated gifs from the generated pictures. But with this approach, it would not be possible to run VisClim on a normal workstation without a very fast processor and without a lot of memory. That was the reason that we brought the JMF (Java Media Framework) into the project. The next milestone was to generate QuckTime movies with the aid of the JMF. We reached that goal at the beginning of August 2003.

An important part of the whole process were the regular demo sessions with the customer. We met all two to three weeks for showing the new functionalities and for getting feedback. These short iterations were helpful for us, because we almost instantly knew if we fulfil the expectations of the customer.

4.2 Personal Impressions

It was a new experience for us to work with so many external resources. We had to read and learn a lot about some foreign frameworks and applications. In a social way, we had to communicate with an external customer who not always thought the same way as we did. We were able to handle these differences because of our frequent meetings.

[1] http://www.cvshome.org/

[2] http://ant.apache.org/

[3] http://maven.apache.org/

In general, we are quite satisfied with the development process and the result. We would not change many things in a next project. Maybe we would extend the time for evaluating the possible technology at the beginning of the project. But according to Tom DeMarco and Timothy Lister, that doesn't matter that much: *The major problems of our work are not so much technological as sociological in nature*. [DeMa 99]

At the moment, we do not really know, what will happen with this application. The people of the Climet group use VisClim right now in order to visualise the data they collected and restored. They also plan to extend VisClim in the future. But there is no one that is officially resposible for the maintenance of that project. If we know more about the future of VisClim, we will publish that on our actual *project homepage*[4]. There are also links to our CVS repository and to some geographical resources.

[4] http://www.iam.unibe.ch/~kobel/project.html

Bibliography

[CARO 03] J. Caron. *NetCDF Java (version 2.1) User's Manual*, 2003. (p 8)

[DEMA 99] T. DeMarco and T. Lister. Peopleware, Productive Projects and Teams. Dorset House, 2nd edition, 1999. (p 19)

[SUN 99] Sun Microsystems, Inc. *Java Media Framework API Guide*, November 1999. (p 7)