# IN505 Systems Analysis



## UML Diagram Suite

Prepared for

**Adrienne Dearnley**

Prepared by

**Adam Charlton, Dave Jackson & Tanvi Ashok**

Otago Polytechnic

2 November 2012

Contents

[UML Class Diagram 1](#_Toc339460700)

[Personnel Involved in Concert 1](#_Toc339460701)

[UML Object Diagram 2](#_Toc339460702)

[Getting Ready for aBand Changeover 2](#_Toc339460703)

[UML Case Diagram 2](#_Toc339460704)

[Preparing to Play Guitar 2](#_Toc339460705)

[UML Activity Diagram 3](#_Toc339460706)

[A Guitar String Breaks On-stage 3](#_Toc339460707)

[UML State Diagram 4](#_Toc339460708)

[Band Behaviour 4](#_Toc339460709)

[Case Tool Evaluation Overview 5](#_Toc339460710)

[Introduction 5](#_Toc339460711)

[Usability of the tool (finding, installing, using, creating output) 5](#_Toc339460712)

[Aspects of CASE tool 5](#_Toc339460713)

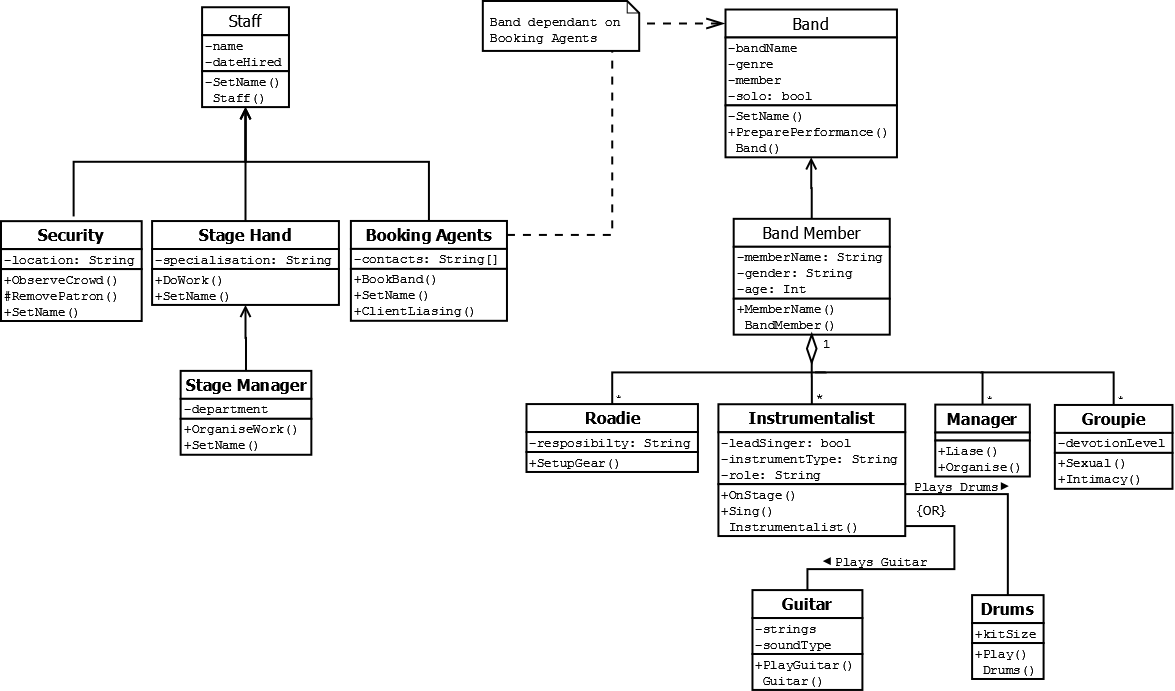
[Reliability and Robustness of CASE tool 5](#_Toc339460714)

[Features of CASE tool 5](#_Toc339460715)

[Opinions of the usefulness of the tool 6](#_Toc339460716)

## UML Class Diagram

### Personnel Involved in Concert



**NOTE:** Abstract Parent Classes are none BOLD, Child Classes are BOLD, Polymorphic methods are also not shown correctly but are implemented in the venue staff and band members.

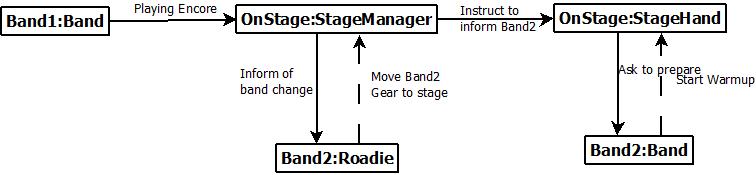
People involved in concert organization include bands and their members (singers, guitarists, bassists, drummers etc.), agents, band managers, stage managers, roadies, security staff etc.

In the above diagram, illustrates the organization of personnel involved in a music concert. This includes venue staff, the band/s and their member’s. All venue classes descends from the Staff class, however; for the band and its member classes to be created. These are dependent on the Booking Agents class and created from the *BookBand()* method.

For example *+BookBand(Band).*

## UML Object Diagram

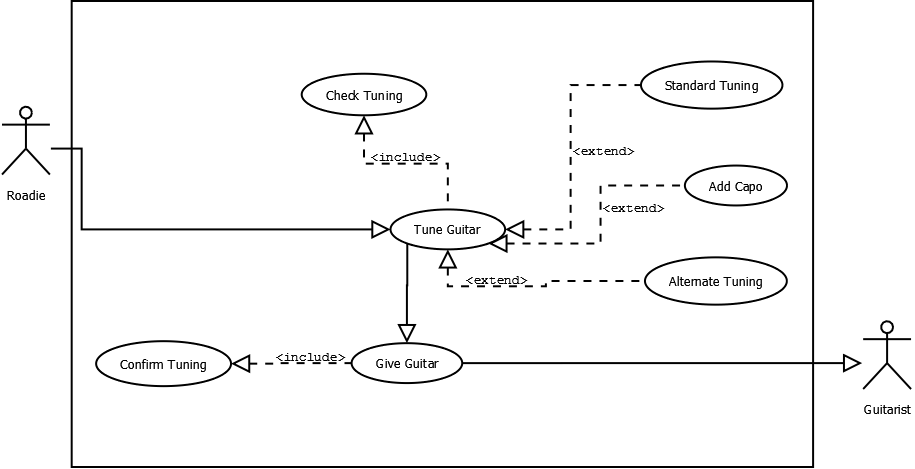
### Getting Ready for a Band Changeover



While Band1 is playing their encore synchronously the OnStage:StageManager asynchronously informs waits for the Roadie to prepare Band2 gear to the stage, the OnStage:StageHand is then called synchronously and then sends an asynchronous message for the band to warm-up.

## UML Case Diagram

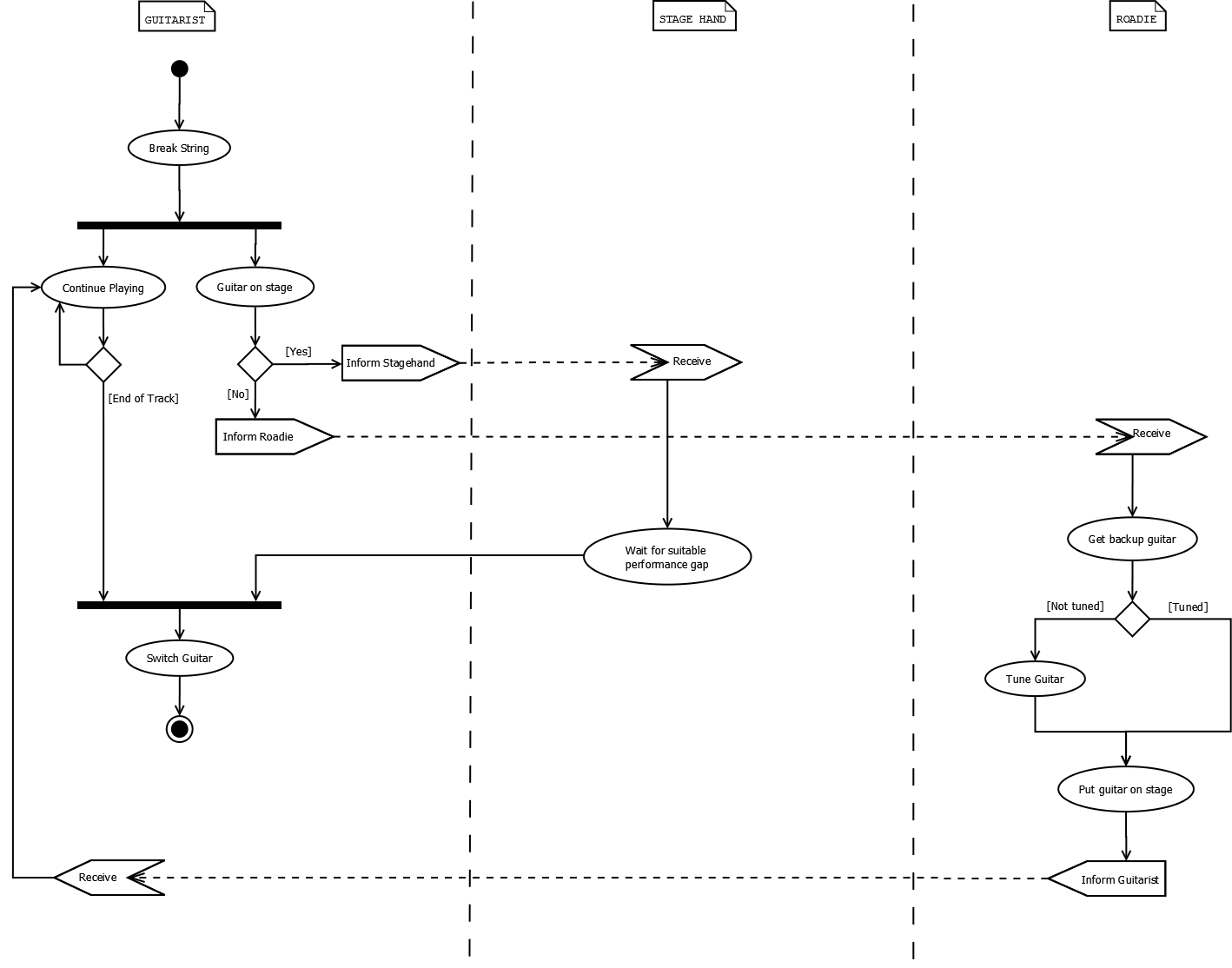
### Preparing to Play Guitar



In this case, the roadie checks the alternate tuning required by looking beforehand at the song-sheet, and confirms the tuning by name to the guitarist as he hands it over.

## UML Activity Diagram

### A Guitar String Breaks On-stage



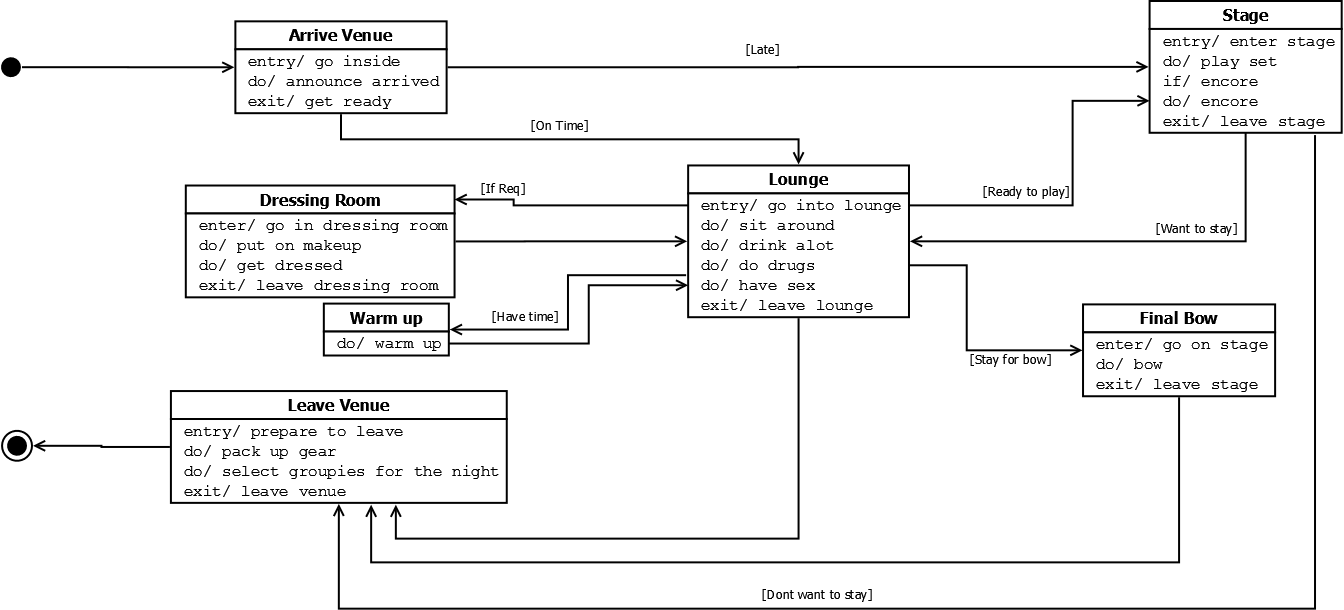
**NOTE:** Limitations of DIA, the boxes used when shifting control to another process are not exact format.

If a guitarist breaks a string whilst playing on stage, they continue playing as best they can. If they have another suitable guitar on stage, they await a suitable gap in their performance, and then signal to a stage-hand to plug it in and bring it to them.

If there is no suitable guitar on stage, then as soon as the string breaks, the guitarist signals to a roadie to: collect a backup guitar from back-stage, check that it is tuned, and bring it on-stage at the end of the track currently being played.

## UML State Diagram

### Band Behaviour



Each band will arrive at the concert venue and is hosted in the lounge, unless they arrive late and are due on stage immediately. Some bands (e.g. glam rock, heavy metal) require time in the dressing room before performing. Before going on stage, bands are given ten minutes warm-up time if possible.

Once on stage, the band plays for an agreed set time or longer. At the end of their set they come off, and then optionally go on again for an encore. After their performance is finished, the band either returns to the lounge or depart. If they remain in the lounge at the end of the concert, they might be invited back on stage for a final bow. After this, the band leaves the venue.

## Case Tool Evaluation Overview

### Introduction

Our task as a group was to select, download and evaluate a CASE tool which would be able to draw the diagrams that we had learnt as part of our course. We were to use this CASE tool diagram to come up with a set of UML diagrams for a Rock Band Concert. However, after reviewing tools, such as: *ArgoUML, Visio, Sparxs, RSM, Visual Paradigm, Altova, StarUML, UMLet and DIA*. The tool that our group decided on was the open source program called “*DIA*”.

Link: [*http://projects.gnome.org/dia/*](http://projects.gnome.org/dia/)

### Usability of the tool (finding, installing, using, creating output)

DIA was fairly easy to find as it was the one that received the most user ratings. As DIA is open source it was easy enough to install on our computer and did not require any user input. It was fairly intuitive to use as everything we needed to create our UML diagrams were right there. The diagrams that it created were to our liking as well.

### Aspects of CASE tool

DIA supports more than 30 different diagram types like flowcharts, network diagrams, database models etc. to enable the user to draw all kinds of professional diagrams.

### Reliability and Robustness of CASE tool

DIA is extremely reliable because of its simplicity. There are no problems installing it and everything is available on the interface. DIA is very useful in connecting tools together and if you drag the arrow it connects one tool to the other.

### Features of CASE tool

* Draw structured diagrams (flowcharts, network layouts, etc)
* Easy to use (Recommended in 89% of the user ratings)
* More than 1000 predefined objects and symbols
* Supports Windows, Mac OS X and Linux
* Many import and export formats
* Scriptable via Python

### Opinions of the usefulness of the tool

I found DIA to be very useful as it was very intuitive compared to the more complicated CASE tools like Visio. Everything that I needed to create my UML diagrams was available in a little drop down menu. I got frustrated trying to use the connecting arrows tool as it was difficult, trying to try and get it working with a touch pad.

I liked the fact that a picture could be saved in a format that was easy to upload. Although there were parts of DIA that did not work the way I wanted it to, overall it did a good job of the diagrams. It has also not crashed since I’ve started using it which is always positive. I give my thumbs up for other professionals who might be looking to create diagrams*. -Tanvi*

DIA is a wonderful tool for creating UML diagrams. As a novice user DIA was very intuitive to learn and use. The interface is self-explanatory allowing the user to easily access the tools required. Scalability of the software works great and is simple enough for a new user but has great depth to the software, becoming an effective tool for any industry professional.

The time spent creating the diagrams were reduced significantly compared to other main stream tools, such as Visio; however, DIA gave me the ability to customize the diagrams without having to really go looking for the options.

On this basis, I would definitely recommend DIA as a tool for creating all types of diagrams for IT professionals. -*Adam*

I found that DIA was the best UML Case Diagram tool to use. DIA was very easy to use and made it simple to do many different types of diagrams. The software isn’t only limited to UML which means you can mix and match different types of diagrams to get the perfect output that is required.

Some limitations I found about DIA was the inability to connect arrows and lines to any part of an object which led to annoying workarounds and the fact that it took a couple of menus to edit

Overall I found that despite its few shortcomings DIA is one of the best diagram suites I have come cross and can easily outperform the expensive suites that claim to offer the ability to what DIA does well. -*David*