Unit 7
Writing and delivering a conclusion, reflecting on difficulties and limitations of a project

Overview

A conclusion is not merely a descriptive summary of your work, but your last opportunity to convince and create an impression for your audience. In this unit, you will learn the purposes, structure, the language features of a conclusion, and the differences between it and an abstract / summary, and an introduction. You will also learn how to reflect on the difficulties and limitations encountered in your project while justifying your decisions and recommending future work.

Learning Outcomes

By the end of this unit, you will be able to

- identify the purpose, main components and language features of a conclusion
- highlight the differences between a conclusion and a summary/abstract/introduction
- draft a summary/abstract for your project report
- articulate the difficulties and limitations of a project

7.1 Purpose of a conclusion

The conclusion is the last section of a report and comes before the list of references, though it is sometimes combined with and included within the section labelled ‘Discussion’. It relates to the aims of the project stated in the introduction and sums up the essential components of the report. It is usually two to three paragraphs in length in a long report. Writing an effective conclusion enhances the overall quality of the report and exerts a significant impact on the audience. It also helps to bridge your ideas about the topic to the readers’ and shows why your findings and analysis should matter to them. As a rule of thumb, the conclusion should:

- state if you have accomplished your aims
- provide a brief summary of the major findings and results in your report
- highlight the chief outcomes and the significance of your investigation
- avoid introducing new information
- provide recommendations or needs for future action (which should be related to the limitations of the current work)

7.1.1 Structure of a conclusion

A conclusion generally features the following five components.

1. **Context**
   Begin your conclusion with restating the objectives of your report, i.e., 'Why did you engage in the investigation?' and 'What did you want to find out?' Then move on to summarize the major actions taken to achieve the objectives. For example, 'What did you do?' and 'How did you do it?

2. **Comments on findings**
   Summarize the major results that address the objectives, i.e. 'What did you find out? You should also provide overall comments on the major results. For example, 'How did you interpret the results?' or 'What is your understanding of the results? Qualitative comments will suffice on most occasions.

3. **Significance/Contributions**
   Discuss the contribution of your investigation, i.e., 'How has your investigation solved/not solved the problem you raised' and 'why?' and 'Does your study offer any insight into the topic?

4. **Limitations**
   State the limitations of your study and acknowledge any remaining problems. For example, 'What remaining "gaps" of knowledge of your topic did you not fill in the investigation?' Or 'Can the approach/method/sample size be improved in future?' 'What's the main weakness of this study?'

5. **Recommendations/Implications – DO NOT FORGET THIS!**
   Suggest any recommendations for further research, i.e., 'How could other researchers build on what you have done?' 'What other changes need to be made for future research?'

TASK 7.1 Identify the main components in a conclusion section

Below is the conclusion of a technical report titled “Example-Centre Programming: Integrating Web Search into the Development Environment.” Read the text and answer the questions below.

1. Identify the main components in the conclusion of a technical report titled “Example-Centre Programming: Integrating Web Search into the Development Environment.”

2. Which components has the writer emphasized more than the others? Why?

3. How much detail does the writer give about the results and discussion section? Why?

Text 1 [1]

	Components
We have presented a user interface for accessing online example code from *within the development environment*. This interface displays search results in an *example-centric manner* to support programming by example modification. This paper described the implementation of *Blueprint*, a lightweight method for using a general-purpose search engine to create code-specific search results that include written descriptions and running examples. Empirical results suggest that *Blueprint*’s approach of integrating web search into the development environment helps programmers acquire and adapt online resources more efficiently. An important avenue for future work is to improve the modification of example code. Copied code can introduce bugs when programmers assume that sample code works and forget to adapt portions of the example. *Blueprint* users would benefit from rich refactoring support for pasted code. This would help users change variable names consistently and reduce the number of errors. It might be valuable to rethink the character-at-a-time editing paradigm entirely. Would it be more efficient to navigate pasted code a token at a time? Perhaps arrow keys should move the user’s cursor between tokens, and typing over top of an existing token should automatically replace all occurrences of that token within the pasted region. While example-centric development is common, there is little aggregated knowledge about how users adapt examples. If *Blueprint* could show users how code has been changed in the past, perhaps they will make fewer errors. For example, if all ten previous users changed a literal, it is highly likely that the eleventh user should change this literal as well. The wisdom of the crowds may enable significant advances in online programming tools.	

7.1.2 Language of a conclusion

A conclusion reports outcomes of a study, but also points to the future as part of the process of refining engineering knowledge. Thus, it points backward highlighting the study and its significance, but also forward to further work. Because of the retrospective and prospective nature of the conclusion, different language patterns and verb choices are required. They have to be selected carefully to give the reader a clear understanding of how any one particular report fits into ongoing developments in the field. Hence, this section focuses on tense choices and how to make claims with the appropriate degree of certainty as illustrated in Table 1.

1. Tense

Components	**General/Dominant Tense***
Context	Varies dependent on the situation
Findings	Simple Past Present Simple (for comments on findings)
Significance / Contributions / Existing situation / General truths & general phenomena	Simple Present, and verbs indicating caution (e.g. may, might, appears, seems, is likely, and is possible, etc.)
Limitations encountered in the project / research	Simple Past (may also include words indicating degree of certainty)
Recommendations/Implications	Simple Present, and common constructions such as 'should/could/would/must be + ed, or 'X is recommended', etc.)

Table 1. Tenses in a conclusion

2. Evaluative vocabulary (see Task 5.4 in Unit 5)

   In addition, highly evaluative vocabulary and expressions such as "this helps programmers acquire and adapt online resources more efficiently" are expected to accompany all the description of past results and recommendations of future work.

3. Hedging vocabulary

   To enhance the receptiveness of your claims, recommendation, and future predictions, hedging devices are required such as (see Unit 5) may, might, appears, seems, is likely, and is possible, etc.

TASK 7.2 Identify the language characteristics of a conclusion

Step 1

Identify the language characteristics of a conclusion of a technical report titled "Example-Centre Programming: Integrating Web Search into the Development Environment."

1. Comment on why each tense / verb form has been used (see Table 1 above). Can any other tenses / verb forms be used instead?
2. There are several modal verb forms used. Comment on why each modal is used. (e.g., suggestion, recommendation, hypothetical situation, something is possible (but not definite), prediction.

Present tense **modal verb** **past tense** **present perfect** **other structures**

Text 2 [1]

	Use of tense/verb form
We **have presented** a user interface for accessing online example code from *within the development environment*. This interface **displays** search results in an *example-centric manner* to support programming by example modification.	Achievement of paper Fact / contribution
This paper **described** the implementation of *Blueprint*, a lightweight method for using a general-purpose search engine to create code-specific search results that include written descriptions and running examples. Empirical results **suggest** that *Blueprint*'s approach of integrating web search into the development environment helps programmers acquire and adapt online resources more efficiently.	
An important avenue for future work **is** to improve the modification of example code. Copied code **can** introduce bugs when programmers **assume** that sample code works and **forget** to adapt portions of the example. *Blueprint* users **would** benefit from rich refactoring support for pasted code. This **would** help users change variable names consistently and reduce the number of errors. **It might be valuable to** rethink the character-at-a-time editing paradigm entirely. **Would** it be more efficient to navigate	

pasted code a token at a time? Perhaps arrow keys **should** move the user's cursor between tokens, and typing over top of an existing token **should** automatically replace all occurrences of that token within the pasted region. While example-centric development **is** common, **there is** little aggregated knowledge about how users **adapt** examples. **If *Blueprint*** **could** **show users how code** **has been changed** **in the past**, perhaps they **will** make fewer errors. For example, **if all ten previous users changed** a literal, **it is highly likely that** the eleventh user **would** change this literal as well. The wisdom of the crowds **may** enable significant advances in online programming tools.	

Step 2

Think about the balance of the different components in the above conclusion.

1. How much of the conclusion is devoted to reporting the main results of the research? Why?

2. How much of the conclusion is devoted to reporting on the implications of the work and recommendations for future work? Why?

3. What are the limitations of *Blueprint* (and similar user interfaces) reported in this research? What solutions / recommendations does the writer suggest solutions to address these? You can mark these up in the extract from Text 2 below.

Extract from Text 2 [1]

	Limitations /Solutions /Recommendations
An important avenue for future work is to improve the modification of example code. Copied code can introduce bugs when programmers assume that sample code works and forget to adapt portions of the example**.** *Blueprint* users would benefit from rich refactoring support for pasted code. This would help users change variable names consistently and reduce the number of errors. It might be valuable to rethink the character-at-a-time	

editing paradigm entirely. Would it be more efficient to navigate pasted code a token at a time? Perhaps arrow keys should move the user's cursor between tokens, and typing over top of an existing token should automatically replace all occurrences of that token within the pasted region. While example-centric development is common, there is little aggregated knowledge about how users adapt examples. If *Blueprint* could show users how code has been changed in the past, perhaps they will make fewer errors. For example, if all ten previous users changed a literal, it is highly likely that the eleventh user would change this literal as well. The wisdom of the crowds may enable significant advances in online programming tools.	

7.2 Writing an abstract/ summary

Similar to a conclusion, an abstract/summary is a compact description of the entire project or research summarizing the major and most important related information.

7.2.1 Differences between an abstract/summary and a conclusion

The best way to identify the purpose of an abstract/summary the differences between the two is to make observations by yourself rather than reading another long text of formal definition.

TASK 7.3 Recognize the differences between an abstract/summary and a conclusion

Read the abstract/summary of the same report (see *Text 3* below) to and compare with the conclusion in the previous task (*Text 2*). Complete the table below:

Feature	Abstract / Summary	Conclusion
Position in a report		
Length	150 – 250 words regardless of the length of the work/research.	Varies, but generally 300 words or longer in a typical student final year project.
Main purpose		
Relative weight of a component (e.g., main findings, recommendations) • Same importance for each? • Some components more important than others?		

Text 3 [1]

The ready availability of online resource code examples has changed the cost structure of programming by example modification. However, current search tools are entirely separate from editing tools. What benefits might be realized by integrating them? This paper describes the design, implementation, and evaluation of *Blueprint*, a tool that integrates Web search into Adobe Flex Builder development environment. *Blueprint* automatically augments queries with code context, presents an example-centric view of search results, and retains a link between copied code and its source. This paper introduces a technique for retrieving relevant example code, descriptions, and running examples for a user's query. A between-subjects study found that *Blueprint* enables participants to search for and select example code significantly faster than with a standard Web browser.

TASK 7.4 Identify the differences between an abstract/summary, a conclusion, and an introduction

Now that you know more about the differences between an abstract/summary and a conclusion, how about comparing them with an introduction (See Unit 3). What are the differences?

7.2.2 The structure of an abstract/summary

There are four main components in an abstract/summary:

1. **Problem/motivation/context/objective**
   The first few sentences set out the context or problem which motivates the work. The opening should raise interest or point out the significance of what is to follow. The purpose or objective may be stated directly. The opening concisely addresses questions, such as, "Why should we care about this?" or "What is the theoretical or practical importance of the work?"

2. **Methodology/procedure/approach**
   This describes how the work was carried out and should provide some significant specific details. This section may refer to specific processes, numerical simulations, construction of prototypes or experimental testing. It addresses the question: "What did you do?"

3. **Results/findings/products/outcomes**
   Objective (mostly quantitative) results are highlighted. The most central outcomes, such as, answers to problems, research questions or project objectives, are featured precisely and without much commentary. The question here is, "As a result of what you did (2), what has been learned, discovered, or created?

4. **Conclusions/recommendations/applications/implications**
   Some interpretations or conclusions are given which focus attention on the value of the work completed. These may be theoretical or practical in nature. The focus is on only a few key points. The question: "So what are the implications or applications?" "Where should research and development go from here?

TASK 7.5 Identify the four moves of an abstract/summary

Below is an abstract/summary of a report by a previous student on an online event registration system. Identify the four main components.

Text 4 [2]

	Components
As there is currently no generic online event registration system, thousands of conference organizers build their own individual registration websites which adds significantly to the cost of conference arrangements. Therefore, there is need for a generic online registration system which can provide all the fundamental functionalities of a conference registration site at a relatively low cost. In this project, a model of a generic online registration system with mail account was investigated and a web application based on this model was developed. The web application was developed using ASP.NET 2.0 with the programming language of C#. By using the Simple Mail Transfer Protocol and the Post Office Protocol, the application is able to connect to the Gmail Server to send or receive emails. With these email functions, the application can embed the participants' information in an email and send it to the Gmail account provided by the conference organizer, and the information stored in the email account can also be retrieved by the receiving function. This application can provide all the fundamental functionalities that conference registration sites need, and it can build the registration database using the mail account provided by the conference organizer. This would lower the cost borne by the conference organizer. The overall performance of this online registration system proved to be generally acceptable, though occasional instability/delays due to network congestion were observed. It is expected that this problem can be overcome with an increase in network bandwidth.	

7.2.3 The language of abstract/summary: Focus on Tense

Tense varies based on the move described above and guidelines and suggestions are provided as follows:

Abstract/Summary Components	Tense
1. Problem/motivation/context/objective	Present - unless historical fact
2. Methodology/procedure/approach	Past
3. Results/findings/products/outcomes	Present / Past
4. Conclusions/recommendations/applications/implications	Present / will + base form

Present Simple:

- a broad statement about the problem or motivation behind the work
- general truths (common knowledge or belief in the field)
- to make generalisations from the data (results and conclusions)
- to refer to what a paper does, how it is organised, what it contains (as in 2 above)

Simple past:

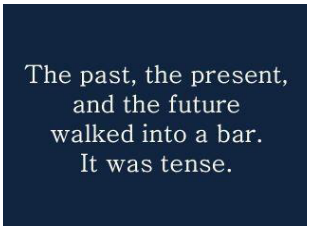
- to describe what you did in your study (methodology)
- to describe what you found (findings)

Present Perfect:

- to summarise what has previously been presented/done
- to refer to whole areas of enquiry

Simple future:

- to refer to applications, recommendations, and implications


The past, the present, and the future walked into a bar. It was tense.

[1]

TASK 7.6 Write an abstract/summary of your project

Write down a list of major items of your project up to this stage, e.g., motivation, scope, major interim results, status, etc. You have to be SUCCINCT. Show it to another classmate who does not know anything about your project. Ask him/her whether your list is significant enough to make him/her read the rest of the report.

7.3 Articulating difficulties and limitations encountered

Reflecting on problems encountered in the project is frequently overlooked by students according to their supervisors. An important value of conducting a project in school is to have learners improve the problem-solving skills for future work. To achieve that, students should be able to articulate, evaluate, and address the problems. This is perhaps even more important than merely reporting reasonable findings and achieving the expected outcomes in the learning process.

Since this section is not as structured as an introduction or a conclusion, reading a variety of such sections in previous student reports will shed light on the ideas to leverage and pitfalls to avoid in your own report.

TASK 7.7 Critique writing on limitations and difficulties encountered

Read the section of "limitations/difficulties encountered" in two reports by previous students. Apply what you have learnt in justification in terms of being evaluative and critical (see Unit 5) and identify the good points to follow and pitfalls to avoid. Use the guiding questions below.

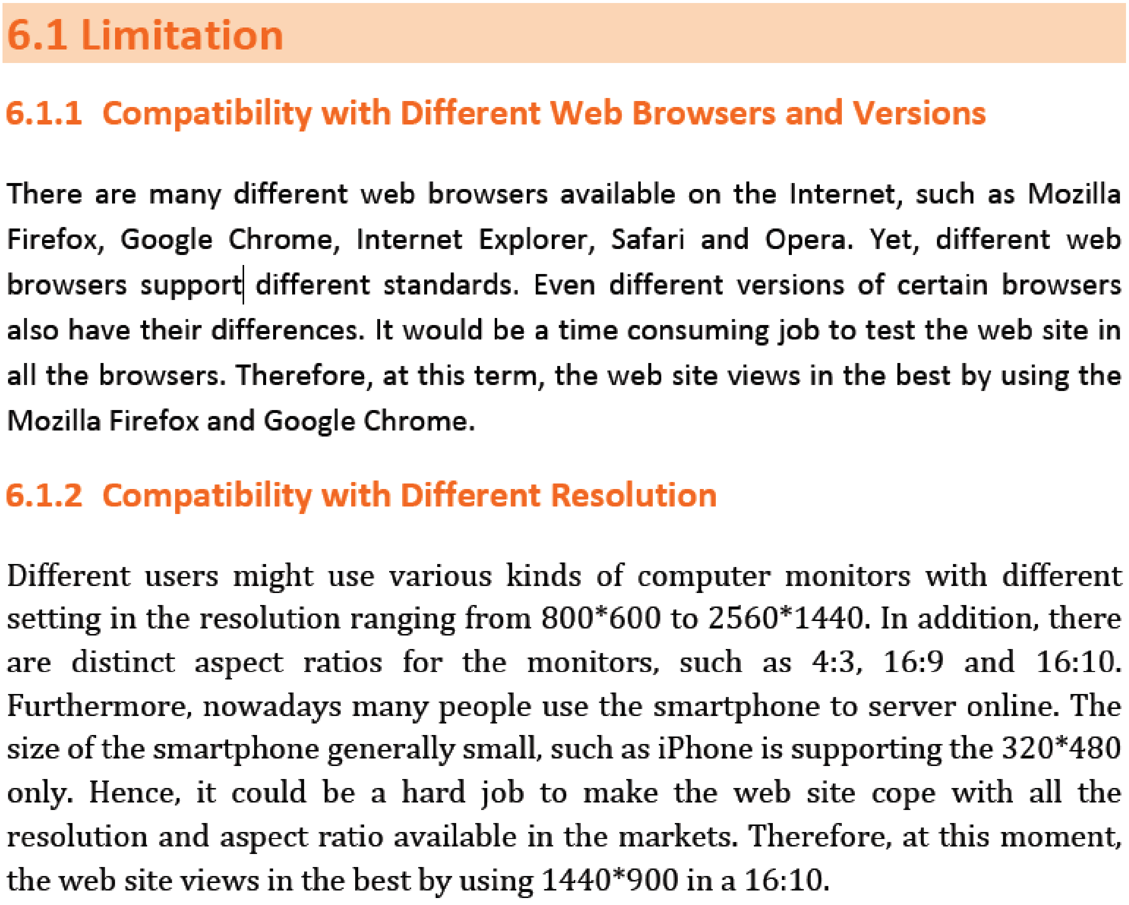
Put yourself in the position of a reader

- Can you understand the difficulty? Can you visualize it?
- Can you assess the **impact and significance** of the difficulty?
- To what extent has the writer overcome the difficulties? How? Any **justification offered if the difficulties are not to be solved**?
- Do these difficulties lead to future work?

Text 5 [2]

Report 1 (Note: In this report, there is not a section with a title "Difficulties encountered")

Title: Where2cut: a Salon & Hairstyle Guide Web Site



6.1 Limitation

6.1.1 Compatibility with Different Web Browsers and Versions

There are many different web browsers available on the Internet, such as Mozilla Firefox, Google Chrome, Internet Explorer, Safari and Opera. Yet, different web browsers support different standards. Even different versions of certain browsers also have their differences. It would be a time consuming job to test the web site in all the browsers. Therefore, at this term, the web site views in the best by using the Mozilla Firefox and Google Chrome.

6.1.2 Compatibility with Different Resolution

Different users might use various kinds of computer monitors with different setting in the resolution ranging from 800*600 to 2560*1440. In addition, there are distinct aspect ratios for the monitors, such as 4:3, 16:9 and 16:10. Furthermore, nowadays many people use the smartphone to server online. The size of the smartphone generally small, such as iPhone is supporting the 320*480 only. Hence, it could be a hard job to make the web site cope with all the resolution and aspect ratio available in the markets. Therefore, at this moment, the web site views in the best by using 1440*900 in a 16:10.

Text 6 [2]

Report 2
Title: Gesture recognition on smart phones

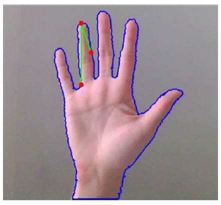


Figure 1. Contour mapping in gesture recognition

10. Future Work

10.1. Adaptive Skin Detection

Currently, the skin detector performs poorly when there is significant change of lighting conditions. This is due to the fact that the parameters are fixed and are not adaptive to the environment. In fact, after we get the detected skin mask, we can update the SPM by calculating the skin-color histogram and all-color histogram online. As long as the lighting condition changes gradually, this method should give certain extent of robustness against different illumination.

Argyros *et al* suggested that by using a weight average of the $P(s|c)$ trained offline from the training set and the online computed one, we can obtain a posterior probability that reflects the recent appearance of skin-colored objects[3].

In fact, the appearance of the head also gives us a hint of the skin tone under the current light sources. We can compute the histograms using pixel on both the hand and face to generate a more suitable SPM. There exists a number of face detection algorithms. If we incorporate these face classifier into our system, we believe the detection rate will be much better. However, we don't know the cost of introducing another task running real task. It may be a huge burden for handheld devices. All these need to be searched in the future.

10.2. Tracking with Momentary Occlusion

In the current system, once the hand blob is overlapped with another blob, the "further one" is removed from the memory. However, sometimes when the hand is performing the swipe gesture, the hand may momentarily occlude the head, and the reappear as a new blob. With Kalman filter, we can actually allow a blob to be lost for a few frames and estimate the position of the hand by prediction without correction before deleting it from the memory.

7.4 Work Accomplished to Date, Work Remaining, and Plans for Next Reporting Period

For an interim report in which substantial results of the study may not be available, a list of items can also be provided to show:

- work accomplished to date
- difficulties encountered with impact and possible resolutions
- work remaining
- plans for the next reporting period with a timeline.

7.5 Over to you

The main theme of this unit is to create an impression, whether it is in the conclusion or in the poster presentation. Apart from the mechanics of structure and language features, you incorporate a 'human' side of creating an 'impact' on the reader by 'pitching' the benefits of your work and articulating the difficulties encountered. This allows your audience, mostly your course lecturer or supervisor, to appreciate how you think and learn.

TASK 7.8 Reflect on this unit

Key points to remember

- The conclusion is your last opportunity to convince your audience. List only the most important points in each area introduced
- Revisit the major differences between a conclusion, an abstract/summary, and an introduction. One should not just paraphrase the other.
- Articulating difficulties encountered: make audience understand, visualize, and assess them. Justify how to overcome the problem. All these lead to future study!

References

[1] Brandt J, Dontcheva M, Weskamp M, Klemmer SR. Example-Centric Programming: Integrating Web Search into the Development Environment. Stanford University; 2009.

[2] Adapted from student texts