

A NARRATIVE APPROACH TO COLLABORATIVE WRITING

A business process model

Peter Henderson, Nishadi De Silva

School of Electronics and Computer Science, University of Southampton, Southampton, UK
p.henderson@ecs.soton.ac.uk, nhds03r@ecs.soton.ac.uk

Keywords: Collaborative writing, Business Processes, Narratives, Rhetorical Structure Theory (RST), CSCW

Abstract: Narratives have been used in the past to enhance technical documents such as research proposals by implementing a single-user writing tool called CANS (Computer-Aided Narrative Support). This study has now been extended to collaborative writing (CW); another area that can greatly benefit from a narrative-based writing tool. Before implementing such an asynchronous, multi-user system, however, it was imperative to do a concrete design for it. Therefore, after studying existing CW tools and strategies, a concise business process (BP) model was designed to describe the process of narrative-based CW. This paper introduces narrative-based CW for technical authors, the BP model for it and discusses the benefits of such an implementation on particular areas of research, such as the development of Grid applications.

1 INTRODUCTION

Collaborative writing (CW) is becoming increasingly common; often compulsory in academic and corporate work. There are many software tools that support CW and address the complications inherent in most multi-user applications. However, a common complaint with CW is the lack of coherence between the independently-authored sections in a document. The current practice of CW groups writing to an agreed outline (Alred et al., 2003) is successful but does not address this problem. Developing a narrative instead of an outline and agreeing to use it as a formal structure to the document will resolve the lack of coherence in collaboratively-authored documents.

It was shown previously (De-Silva and Henderson, 2005, De-Silva, 2005) how narratives enhance single-author technical documents. A tool called CANS was built based on Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) that guides an author through a narrative-driven writing process. This research was then extended to build a similar tool for CW.

It is an established fact that successful applications rely on well understood business processes (Henderson, 2000). Therefore, after studying existing CW strategies and tools, a business process model was designed to describe the collaborative processes of reading, writing and reviewing, from a

narrative perspective. This model is currently being implemented using Web Services.

There are diagrammatic ways of presenting business processes in UML (Maciel et al., 2005) which are ideal for complex distributed systems. However, since the focus is mainly on introducing narratives to CW, the business processes remain simple and do not warrant the use of such diagrams. Pseudocode is used instead.

This paper has brief introductions to narratives, RST and CW. Following this, the business process model for narrative-based CW is described and its implementation and uses are discussed, particularly as an application suitable for deployment on the Grid.

2 OVERVIEW OF NARRATIVES

A narrative is a representation of events meaningfully connected in a temporal and causal way (Onega and Landa, 1996, Abbott, 2002). For the scope of this paper, it is sufficient to think of a narrative as being analogous to a story.

A *document narrative* is the implicit ‘story’ a document conveys to the reader; a fundamental aspect of a successful document. A document narrative clarifies the authors’ intentions and provides a coherent structure. Narratologists and

linguists have developed several theories to analyse and synthesise coherent narratives. Rhetorical Structure Theory (RST) was chosen to produce coherent document narratives.

2.1 Rhetorical Structure Theory

When a narrative is analysed using RST, it is divided into text segments and relationships are defined between them. A segment is of arbitrary size; often a clause or sentence. The relationships can be illustrated using diagrams (Figure 1). A text segment assumes one of two roles in a relationship: the nucleus (N) or the satellite (S). Nuclei express what is more essential to the understanding of the narrative. Satellites provide supporting information. However, in this paper, there is no great distinction between nuclei and satellites.

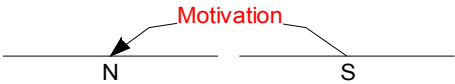


Figure 1: A motivation relationship in RST (N=Nucleus, S=Satellite)

There is an overall effect associated with each relation. For instance, if the nucleus in a MOTIVATION relation presents an action, the satellite should increase the reader’s motivation to perform it. There are 23 relations defined in (Mann and Thompson, 1988). The RST analyses in this paper use just six of these relations (Table 1). RST identifies a hierarchical structure in text. Therefore, a coherent text is expected to produce a tree of relations. It is possible to have multiple valid RST trees for one narrative.

Relation	Description
Background	Provides background information
Elaboration	Provides extra information
Justify	Justifies a certain decision
Motivation	Provides motivation to perform an action
Sequence	Links events that happen in sequence
Solutionhood	Provides a solution to a problem

Table 1: Subset of RST relations used in this paper

2.2 Narrative-based writing

To familiarize the reader with the use of RST on document narratives, an example of writing a fable is presented. Later, these techniques are applied to technical documents. The document narrative below is the structure for a set of fables that an author wishes to write.

I want to write a short story that will contain an implicit moral lesson.¹ I will use animal characters with human features.² I believe this will convey the wisdom in an enjoyable and memorable way.³ I will introduce two or three characters with opposite human characteristics (one righteous, one immoral).⁴ These characteristics will be revealed through brief conversations at the start of the story.⁵ Then there will be a series of events that will be tailored to demonstrate that the characters with the moral attitude always win and that the others suffer consequences for their unwise actions.⁶ Thus the reader will be gently persuaded to take on the characteristics of the successful characters.⁷

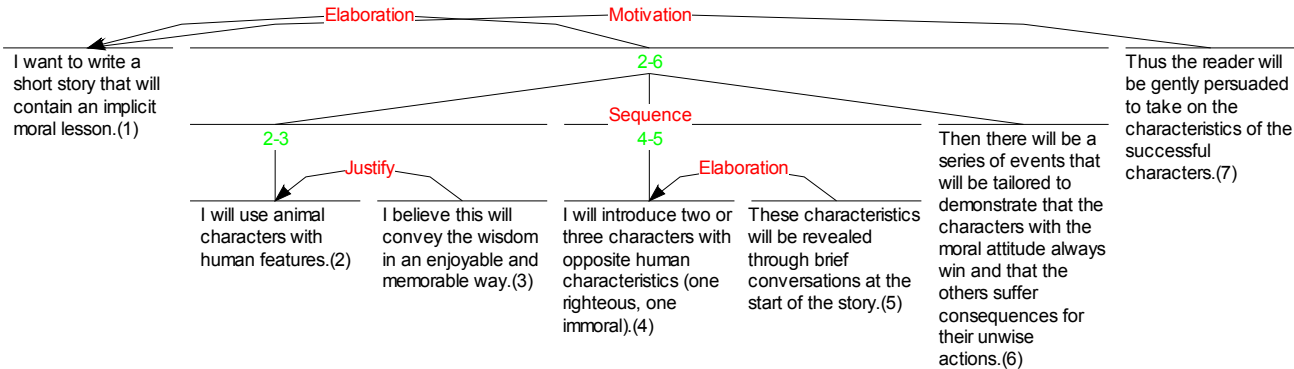


Figure 2: RST analysis of the document narrative for a set of fables

RST is used as a tool to verify coherence. If the document narrative can be placed in a RST tree, it is assumed to be coherent. The document narrative for the fables was analysed using RST (see Figure 2¹). A fable structured according to this narrative should satisfy all the RST relationships. For instance, the famous story of the Ant and the Grasshopper (see Appendix) fits this document narrative. This fable can be divided into segments (see Appendix where the fable is divided into three segments, corresponding to sections 4, 5 and 6 of the narrative above) and each segment creates the expected effect on the reader. Section 4 introduces the Ant and the Grasshopper and section 5 elaborates their characteristics using dialogue. Section 6 presents the onset of winter when the hardworking Ant emerges as the winner. These sections are placed in sequence and convince the reader that the Ant is the better role model. The ‘Motivation’ and ‘Justify’ relations are satisfied too.

This process can be applied to technical documents. See (De-Silva and Henderson, 2005) for a document narrative for a research proposal and section 3.3 of this paper for a generic narrative for a scientific conference presentation.

3 COLLABORATIVE WRITING

Collaborative writing (CW) is the process of multiple authors producing one document, by writing together and soliciting one another’s opinions about their writing.

Since the early 70’s there has been great interest to provide computer support for this process (Noël and Robert, 2004); a move made more concrete by the formation of a specific field of research called Computer-Supported Collaborative Working (CSCW). The resulting groupware and the World Wide Web have revolutionized the art of writing together. In preparation for this paper, CW software such as PREP (Neuwirth et al., 1990) were studied and newer technologies such as Wikis were experimented with (JotSpot, 2004).

¹ All RST trees in this paper have been drawn using the free software tool, RST Tool.
O’DONELL, M. (2000) RSTTool 2.4 – A markup tool for Rhetorical Structure Theory. Proceedings, International Natural Language Generation Conference (INLG’2000). Mitzpe Ramon, Israel.

3.1 Collaborative writing strategies

A CW group usually has an agreed-upon strategy for producing the document which defines how the co-authors will coordinate. There are several strategies to choose from. Two popular methods are discussed below.

3.1.1 Sequential writing model

In this model, only one person writes at a given time and once his/her task is complete, passes the document along to the author next in line. This model is easy to organise and improves coordination between the authors.

However, there are several disadvantages (Lowry et al., 2004) such as the lack of group consensus and the difficulty in ensuring that all document sections are addressed adequately. Also, the *order* of authors greatly influences the final document. One author can change previous contributions or significantly bias subsequent authors.

3.1.2 Parallel writing model

With this strategy, a team divides the CW task into discrete units and works in parallel. This model has several variants. In one, each team member is assigned roles such as ‘writer’, ‘reviewer’, ‘editor’ and so on, depending on their expertise. Members then work on the document according to their roles. In another variation, the document is divided into sections and each author is assigned a section that he/she is responsible for. The completed sections are submitted to the team leader who assembles them together to form the final document. This approach is sometimes called horizontal-division writing (Lowry et al., 2004) and is the model that this paper concentrates on. In (Alred et al., 2003), this process is described in more detail (reproduced below).

1. Designate one person as the team coordinator.
2. Collectively identify the audience, purpose and project scope.
3. Create a working outline of the document.
4. Assign segments or tasks to each team member.
5. Establish a schedule: due dates for drafts, revisions, and final documents.
6. Agree on a standard reference guide for style and format.
7. Research and write drafts of document segments.
8. Exchange segments for team member reviews.
9. Revise segments as needed.
10. Meet your established goals.

(Source: Alred et al., 2003)

3.2 Problems with CW

There are several known problems with CW and three of them are discussed here.

Access control: CW groups have varying access requirements. Most often, only authorized authors are allowed to edit the document. Sometimes fine-grained protection is required; for example, an author may have edit privileges over only one section in the document (Shen and Dewan, 1992).

Version control: If two authors check out version X of the document and, after some period of time, submit edited documents back to the system, there is a chance one version could overwrite the other. Even if both versions are saved, which of the two versions would be the latest?

Lack of coherence: Another drawback of parallel writing is the lack of coherence between the sections that have been independently authored (Lowry et al., 2004). The team leader is often burdened with the task of collating these sections to produce a consistent, coherent document. A narrative approach to CW will help solve this problem.

3.3 Narrative-based CW

Instead of an outline, a CW group can decide on a document narrative. They can also determine narratives for each section of the document. Then authors can craft their respective sections with full understanding of the implicit and explicit narrative goals that the document is expected to fulfil. This will result in improved coherence and reduced work for the team leader.

Authors may also be allowed to change the narrative or RST relations during the writing process. If the CW team agrees on the changes, they will modify the sections to satisfy the new narrative.

As an example of a collaboratively produced technical document, a generic document narrative for a presentation of scientific results at a conference is given below. Through this narrative, the scientists involved in this presentation share and clarify their intentions and agree on one ‘story’ that they wish to convey.

This narrative can be divided among the team and each scientist can construct a part of the presentation such that it satisfies the narrative. For instance, the Previous Work section should convince the audience that it is absolutely necessary to solve this problem and that no one else has been successful so far. It should conclude with a lead to the next section by stating that previous work has, however, helped this team develop their experiments.

There was an unsolved problem in this scientific field and we have solved it.¹ Our research into previous work revealed that there was no complete solution to this particular problem² and this lack was affecting specific groups of people.³ We gathered some useful ideas from these previous researchers⁴ and set about designing our own experiments to overcome the hurdles that they faced.⁵ Here is the design of the experiments we conducted⁶ and a list of our results.⁷ These results are much better than those of our predecessors but we hope to improve them further by conducting more experiments.⁸ Thereby, we conclude that our results are currently the best in this field and greatly help the people who were most affected by this problem.⁹

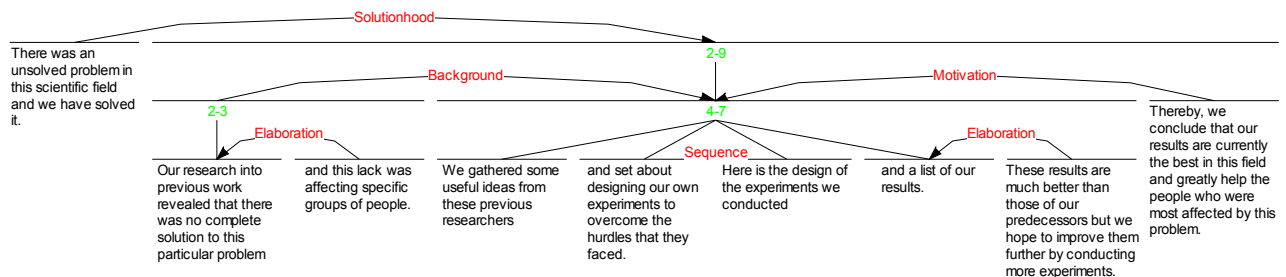


Figure 3: Document narrative and RST tree for a presentation of scientific results at a conference

4 BUSINESS PROCESS MODEL

Narratives are a powerful contribution to CW. Therefore, a narrative-based tool is being developed to support asynchronous CW. Such an application is not easy to design and requires a way of precisely articulating how the authors can interact with the system. Therefore, the business process (BP) model presented here was used to design this tool. In this context, a BP model is an enumeration of all the actions in which the participants can engage. Although it may seem unusual to describe CW as a business, it has all the characteristics of coordinated actions inherent in any business.

To be formal about a BP, these actions, the order in which they can be performed and the effect they have on the shared global state must be stated. This is non-trivial when there is more than one author working asynchronously. Unusually, there is no constraint on the order in which these actions can be performed.

In the model, there is a repository which contains, for now, one document and the corresponding narrative. Multiple authors have access to this repository and can perform a set of actions. These actions are described below using pseudocode.

Every changed document or narrative submitted to the repository is saved as a new version and assigned the next biggest version number. Any of these versions can be retrieved by providing its unique version number. The version with the highest number is assumed to be the most recent. Each version also holds the version number of the document it was derived from (called `parent_ver`). In the following pseudocode, 'p', 'n' and 'd' refer to version numbers.

4.1 Document

Properties of a document

- `ver`: The version number of this document. When a document is first created, it gets version number 0.
- `parent_ver`: The version number of the document that this document is based on.
- `checked_all`: This property is `TRUE` when this document satisfies all the relations in the LATEST narrative. Each version of the document starts with `checked_all=FALSE` because it is assumed that one or more of the sections fail to satisfy the latest narrative. After several cycles of read, write and review, the document should reach the ideal state where

`checked_all=TRUE`. Documents that do not satisfy the latest narrative (but may have satisfied older narratives) are not acceptable. Therefore, when the narrative is modified, the `checked_all` property of all document versions are set to `FALSE` based on the assumption that all documents no longer satisfy the new narrative.

4.2 Narrative

Properties of a narrative

- `ver`: The version number of this narrative. When a narrative is first created, it gets version number 0.
- `parent_ver`: The version number of the narrative that this narrative is based on.

4.3 Repository

Properties and actions relevant to the documents in the repository are prefixed with the letter 'd' and those pertaining to the narratives with 'n'. 'Rep' denotes the repository.

Properties of the repository

- `dlatest`: The version number of the latest document
- `nlatest`: The version number of the latest narrative

Actions that can be performed on the repository

- `dget(v)`: Returns the document with version number `v` ($0 \leq v \leq \text{rep.dlatest}$)
- `nget(v)`: Returns the narrative with version number `v` ($0 \leq v \leq \text{rep.nlatest}$)
- `dput(doc,p)`: Saves the document `doc` in the repository and sets the following properties:

```
doc.parent_ver = p
doc.ver = rep.dlatest + 1
doc.checked_all = FALSE
rep.dlatest = doc.ver
```
- `nput(nar,p)`: Saves the narrative `nar` in the repository and sets the following properties:

```
nar.parent_ver = p
nar.ver = rep.nlatest + 1
rep.nlatest = nar.ver
Set checked_all property of all
document versions to FALSE
```
- `check(doc,nar)`: This is a process performed by authors/reviewers to check if the

sections in document `doc` satisfy the relationships in narrative `nar` (see section 2.2 for an example of a document that satisfies its narrative). 'Nar' is often the latest narrative. If *each* section satisfies all the relationships pertaining to it, then the property `checked_all` of document `doc` is set to `TRUE`.

4.4 Specification of an author's tasks

Typically, a co-author's tasks in this model are reading, writing and reviewing the document and, reading and editing the narrative. These tasks are described below. 'Do forever' loops mean that an action can be repeated as many times as the author wishes until the document is complete.

1. Read latest version of document

```
do forever {
  Document doc;
  doc = rep.dget(rep.dlatest);
  ~~~~ read ~~~~
}
```

2. Edit version *v* of the document and submit new version to the repository.

```
do forever {
  Document doc;
  doc = rep.dget(v);
  ~~~~ edit ~~~~
  doc' = modified doc;
  rep.dput(doc', v);
}
```

3. Review a version of the document

An author or reviewer retrieves a version of the document and checks it against the latest narrative. If the narrative is satisfied, `checked_all` of the document is set to `TRUE`. If any of the narrative goals are not fulfilled, the authors need to edit the document or the narrative so that they once again become fulfilled.

```
do forever {
  Document doc = rep.dget (v);
  Narrative nar= rep.nget (rep.nlatest);
  ~~~~ review content of doc ~~~~
  rep.check (doc,nar);
  if all relationships are satisfied {
    doc.checked_all = TRUE;
  }
}
```

4. Read the latest version of the narrative

```
do forever {
  Narrative nar;
  nar = rep.nget(rep.nlatest);
  ~~~~ read ~~~~
}
```

5. Edit version *v* of the narrative and submit new version to the repository.

```
do forever {
  Narrative nar;
  nar = rep.nget(v);
  ~~~~ edit ~~~~
  nar' = modified nar;
  rep.nput(nar', v);
}
```

4.5 Authorized access and version control

The repository can maintain a table with the author IDs and their access rights. Before each request to read or edit a document/narrative, this table can be queried and only authorized authors can be allowed to perform these operations.

Version control needs to be addressed in more detail but is not the focus of this research. In this model, any previous version can be used to produce a new one. Each new version is stored with a link to the document it was derived from (`parent_ver`), thus producing a tree of versions as shown below.

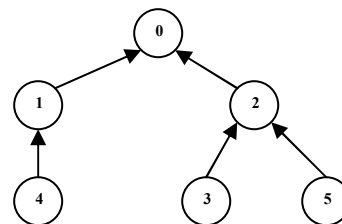


Figure 4: Tree of document versions in the repository

According to this model, version 5 of this document will be considered the latest. However, versions 3, 4 and 5 are at the same level in the tree and have equal chances of being superior or the most appropriate. Either the authors will be given the authority to choose the latest version among themselves or the model will be enhanced to reconcile the differences between the leaves of the tree. This approach with

maximum freedom has been selected over more restrictive methods such as ‘check-in, check-out’ version control because, in designing this creative writing model, the aim has been to support rather than constrain the authors.

5 CONCLUSION AND FUTURE WORK

Collaborating with other authors to write is increasingly common. A document thus produced is often lacking in coherence because of the independently-authored sections. Research presented in this and previous papers (De-Silva and Henderson, 2005, De-Silva, 2005) has shown that applying narratives and RST can significantly improve coherence in single-author and collaboratively-produced documents.

A critical factor for an effective document is a coherent *document narrative*. Noticing a lack of support for document narratives in existing writing tools, we developed CANS (De-Silva and Henderson, 2005): a tool that guides an author through a narrative-based writing process (Figure 5). Ideas from this single-user tool are now being developed into a multi-user application that supports asynchronous CW.

Before embarking on this, it was critical to articulate how multiple authors will interact with the system. This was achieved by designing a business process model for the ‘business’ of narrative-based CW. This model is implemented using a Web Service. The document and all updates to it are stored in an XML database maintained by Xindice (Xindice, 2004).

Finer details of this model are constantly being refined. A more elaborate model will include a record of which documents satisfied previous versions of the narrative. Also, it is possible that some sections in a document will remain unaffected even after the narrative is updated. So, a document can be tracked more closely by assigning each of its sections a ‘checked’ property which is true only if the section satisfies the RST relations in the latest narrative. After a new narrative has been submitted, only sections with `checked=FALSE` will need to be changed.

Owing to the authors’ involvement with the Open Middleware Infrastructure Institute (OMII, 2005) collaborative working over the Grid is of particular interest. Collaborative writing is an ideal application for the Grid. After sufficient development and testing, the narrative-based CW tool will be deployed on the OMII middleware to serve as an example.

During this process, several important lessons can be learnt. Simple and generic ways to convert single-author tools such as CANS to multi-user applications deployable on the Grid will be invaluable to the Grid community. It will also be useful to enumerate the differences between Web-based and Grid-based applications. One such difference is the continuous change in resource locations in the Grid that require location-independent design and implementation. So, documents/narratives in a CW application could move or be split across multiple databases, but authors should still have an interface which gives them reliable access to their work.

Figure 5: Screen shot of single-author writing tool CANS (First step: Entering document narrative)

This work builds an island amidst several strands of parallel research and brings together many technologies. The current CW practice of working to an outline is adequate, but does not solve the lack of coherence. Therefore, narratives are a novel and better approach, providing more support for CW. Document production from a narrative aspect will revolutionize the way people read, write and evaluate documents.

REFERENCES

ABBOTT, H. P. (2002) *The cambridge introduction to narrative*, Cambridge, UK, Cambridge University Press.

ALRED, G. J., BRUSAW, C. T. & OLIU, W. E. (2003) *Handbook of technical writing*, Boston, MA, Bedford/St. Martin's.

DE-SILVA, N. (2005) A narrative approach to technical document construction. *PREP 2005, awarded best oral presentation in Computer Science*. Lancaster University, UK.

DE-SILVA, N. & HENDERSON, P. (2005) Narrative Support for Technical Documents: Formalising Rhetorical Structure Theory. *7th International Conference on Enterprise Information Systems (ICEIS)*. Miami, FL, USA.

HENDERSON, P. (2000) Business Processes, Legacy Systems and a Fully Flexible Future. IN HENDERSON, P. (Ed.) *Systems Engineering for Business Process Change*. New York, USA, Springer-Verlag.

JOTSPOT (2004) JotSpot - The application Wiki. Found at <http://www.jot.com/index.php> (Last accessed June, 2005).

LONG, J. R. (1997) Aesop's fables: Online collection. Found at <http://www.pacificnet.net/~johnr/cgi/aesop1.cgi?sel&TheAntandtheGrasshopper&&antgrass.ram> (Last accessed: 6.10.2005).

LOWRY, P. B., CURTIS, A. & LOWRY, M. R. (2004) Building a taxonomy and nomenclature of collaborative writing to improve interdisciplinary research and practice. *Journal of Business Communication*, 41, 66-99.

MACIEL, R. S. P., SILVA, B. C. D., FERRAZ, C. A. G. & ROSA, N. S. (2005) An MDA-EDOC based development process for distributed applications. *7th International Conference on Enterprise Information Systems (ICEIS)*. Miami, USA.

MANN, W. & THOMPSON, S. (1988) Rhetorical Structure Theory: Toward a functional theory of text organisation. *Text*, 8, 243-281.

NEUWIRTH, C. M., KAUFER, D. S., CHANDHOK, R. & MORRIS, J. H. (1990) Issues in the design of computer support for co-authoring and commenting.

3rd conference on computer-supported cooperative work. LA, California, USA, ACM press.

NOËL, S. & ROBERT, J.-M. (2004) Empirical Study on Collaborative Writing: What do co-authors do, use, and like? *Computer Supported Cooperative Work*, 13.

OMII (2005) <http://www.omii.ac.uk/> (Last accessed: 11.10.2005).

ONEGA, S. & LANDA, J. (1996) Introduction. IN ONEGA, S. & LANDA, J. (Eds.) *Narratology*. New York and London, Longman Group Ltd.

SHEN, H. & DEWAN, P. (1992) Access control for Collaborative Environments. *ACM conference on Computer Supported Cooperative Work*. Toronto, Canada, ACM press.

XINDICE (2004) Online at <http://xml.apache.org/xindice/> (Last accessed on 25.6.2004).

APPENDIX

The Ant and the Grasshopper

In a field one summer's day a Grasshopper was hopping about, chirping and singing to its heart's content. An Ant passed by, bearing along with great toil an ear of corn he was taking to the nest. } 4

"Why not come and chat with me," said the Grasshopper, "instead of toiling and moiling in that way?"

"I am helping to lay up food for the winter," said the Ant, "and recommend you to do the same."

"Why bother about winter?" said the Grasshopper; "we have got plenty of food at present." } 5

But the Ant went on its way and continued its toil. When the winter came the Grasshopper had no food and found itself dying of hunger, while it saw the ants distributing every day corn and grain from the stores they had collected in the summer. } 6

Then the Grasshopper knew: It is best to prepare for the days of necessity.

(Long, 1997)

*Segments numbered according to document narrative in section 2.2