

Article

# Research of Automatic Generation for Engineering Geological Survey Reports Based on a Four-Dimensional Dynamic Template

Yujiao Lei <sup>1</sup>, Jiqui Deng <sup>1,\*</sup>, Jian Lin <sup>2</sup>, Jeffrey M. Dick <sup>1</sup>, Mohammad Naser Lessani <sup>1</sup>  
and Chaoyue Liu <sup>3</sup>

<sup>1</sup> School of Geosciences and Info-Physics, Central South University, Changsha 410083, China; leiyujiao@csu.edu.cn (Y.L.); jeff@chnosz.net (J.M.D.); naser11@csu.edu.cn (M.N.L.)

<sup>2</sup> School of Resources & Environment and Safety Engineering, Hunan University of Science and Technology, Xiangtan 411201, China; 1100018@hnust.edu.cn

<sup>3</sup> Guizhou Traffic Planning, Survey and Design Institute Co., Ltd., Guiyang 550000, China; 185012154@csu.edu.cn

\* Correspondence: csugis@csu.edu.cn; Tel.: +86-138-7495-0729

Received: 10 July 2020; Accepted: 19 August 2020; Published: 21 August 2020



**Abstract:** Errors and inefficiency may be caused by manual processing of complex templates for the preparation and management of engineering survey reports. To address this problem, this paper analyzes the multidimensional variable features of professional field documents and proposes a generation model of standardized reports based on a four-dimensional dynamic template. This approach splits the standardized report into multiple parts to construct a hierarchical tree that represents the report structure according to the report rules, then stores the tree in a graph database, and finally generates the desired report dynamically by retrieving the relational tree for the template and obtaining the relevant data. The model has been applied to the engineering geological survey report system and is shown to improve the working efficiency and data accuracy of the report preparation. Results indicate that the model is feasible and effective.

**Keywords:** report generation; four-dimensional dynamic template; template customization; template hierarchical tree

## 1. Introduction

In the process of engineering geological surveys and management, a large number of technical reports are written and used. The engineering geological survey report is the final result of the survey and is a crucial basis for related engineering design and construction [1,2]. The existence of different report templates in different fields, different industries, different subjects, different organizations and different stages makes it time-consuming and laborious for human beings to deal with complicated templates that have various requirements, and it is impossible to ensure that there are no omissions or errors. It has become an urgent demand to effectively manage and automatically prepare engineering reports through automatic report generation systems. The realization of these systems needs to first develop the key technology of automatic generation of engineering geological survey reports, such as template customization, template storage, data filling, and programmatic generation.

The engineering geological survey report that can automatically be computer generated is a kind of professional field document with strong standardization. Such documents already exist in various application fields and are usually composed of relatively fixed multiple text fields with various expression functions, and related field knowledge. All of these standardized reports have

strong preciseness and a scientific nature, such as product design specifications [3], flood control documents [4], and evaluation documents [5].

Automatic report generation technology belongs to the category of natural language generation (NLG). It is obvious that NLG has become a worldwide research hotspot in the field of artificial intelligence [6]. Its examples include text-to-text generation and data-to-text generation. The production of engineering geological survey reports is a kind of data to text generation. NLG uses the methods of statistics [7], graph sorting [8], machine learning [9], and deep learning [10] to recognize and transform data such as text, voice, and image; then to extract knowledge or logical relationship; and then to generate natural language from machine expression system such as knowledge base or logical form. It can help people better understand the data, generate personalized content on a large scale and accelerate content generation. Its typical applications include artificial intelligence news writing [11], chatbots [12], and business intelligence interpretation and report generation [13]. The text generated based on artificial intelligence, however, has certain randomness, and some problems such as uncontrollable content, redundancy, and weak logic, which lead to insufficient standardization of documents. It is difficult for intelligent methods to meet preciseness and standardization requirements as effectively as those generated using current methodologies.

Moreover, the document generation method based on template [14–26] is widely used for automatic generation of various reports. It requires a pre-customized report template, the program reads dynamic data from the database and automatically fills in the corresponding position in the template to generate the report, which can reduce the amount of code and improve the efficiency of implementation [26]. However, the structure and content of the template are mostly fixed. The method not only prevents adjustment based on data changes and data interaction after the template modifications, but also fails to enable the outline structure, paragraph text, and format details of the document template to change with data sources.

In this paper, we build a standardized report automatic generation model based on a four-dimensional dynamic template by analyzing the multidimensional variable features of professional field documents, extracting the content of different dimensions of the variable template, and choosing the different storage forms of the template. The model is verified by the generation of actual engineering geological survey reports. The experimental results show that the model can solve the problems of chapters, paragraphs, and format details being flexibly changed and adjusted according to dynamic data changes. In addition, it can well meet the needs of engineering survey report compilation and management and improve work efficiency and data accuracy.

The remainder of this paper is organized as follows. Section 2 provides an overview of some related works. Section 3 shows variable features of reports and the four-dimensional dynamic template. Section 4 presents the method and technology. Section 5 discusses the experimental data and results. Section 6 concludes this paper and provides suggestions for future research.

## 2. Related Work

Studies [14–26] address the challenge of meeting rigorous reporting requirements of automatically generated professional field documents by utilizing a template that is customized in advance. When the report is generated, the program reads the dynamic data from the database and automatically fills in the corresponding position in the template. Studies [14–22] employed the method of finding and replacing bookmarks to insert numerical values and graphs into a pre-established report template. The advantage is that the style of the document can be designed outside the program. The amount of code is reduced, and the execution efficiency is improved, but it brings great inconvenience to future program upgrades and maintenance. Dong [23] utilized the function of Word tables to fill the table in the Word report template cell by cell. In addition, domain variables were used to generate content such as text and graphs in the template and to control its format. This method can only generate simple and fixed format charts and is not suitable for generating complex dynamic charts. Liu [24] encapsulated multiple data processing methods into data placeholders and inserted them into Word

documents to customize report templates, and then used report templates to generate Aspx dynamic web pages. Finally, the corresponding data processing methods were invoked to input or retrieve information in the light of the user operation and replaced the data placeholders to generate the final report. Qu [25] proposed a mathematical model of automatic document generation that defines some necessary operations and document templates required in the process of document generation. Then, it converts the template to a data summary document with a specific date by means of the iterative calculation of the model. This method can well meet users' needs for documented data analysis methods. However, the document template format and content need to be defined in advance, which is not conducive to future modifications and improvements.

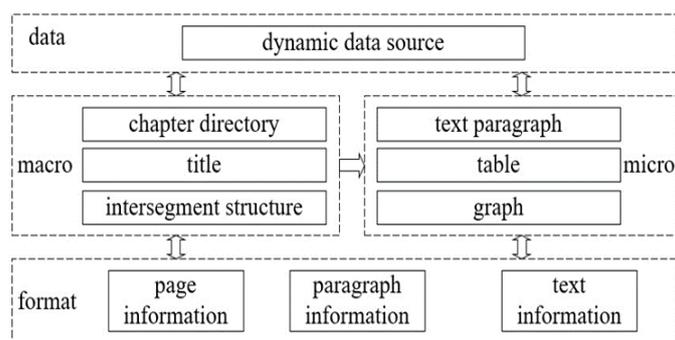
### 3. Variable Features of Reports and the Four-Dimensional Dynamic Template

#### 3.1. Types of Variable Features of Reports

From the perspective of the presentation a report, the report consists of text, graphs, and forms. From the perspective of report content, the report contains cover, catalog, body, and back cover. The main body is composed of multiple paragraphs, which may include text, formulas, graphs, and tables. There are certain relationships between paragraphs, such as causality, explanation, and parallelism. The in-segment text is divided into fixed text and dynamic text; the latter is closely related to the data source. Formulas are expressed by OLE (Object Linking and Embedding) objects, pictures, and characters. Charts contain information such as number and title. A report can contain multiple format information, such as text in different locations or different types of text in the same location. A change of data source may cause changes to structure of the outline, paragraph content, and corresponding format information. Therefore, we can represent the content and form of report in four dimensions: outline, paragraph, data, and format.

#### 3.2. Four-Dimensional Dynamic Template (FDDT)

A template is a special document that provides the basic tools and text to shape the appearance of the final document. Different templates are adopted according to different applications in order to improve work efficiency and achieve completely automated operation. Defining the content and form of a template is the basis of document generation, and we can define the corresponding four-dimensional dynamic template according to the different types of variable features of the report, as shown in Figure 1.



**Figure 1.** Structure of the four-dimensional dynamic template.

By considering the content and form of the report from the perspectives of macro, micro, data, and format, the template can be divided into four parts: the outline structure, the paragraph content, the dynamic data, and the format. The outline structure responds to dynamic data. There are many branches in the paragraph structures, which are selected according to different data conditions.

The formatting, which includes font, font size, and color, is adjusted to produce a standardized report based on the purpose of the report.

According to the report requirements, we divide a standardized report into numerous parts so as to construct a hierarchical tree of the report structure. We store and manage the tree in a graph database, then acquire the tree data through traversing the tree and store this data in a relational database, and finally establish the relationship between the tree and dynamic data sources. When generating the report, the stored procedure generates the relationship tree for a template and acquires relevant data through the retrieval conditions and converts the integrated data into the LaTeX format to generate a final report.

### 3.3. Four-Dimensional Dynamic Model

The four-dimensional dynamic model is a hierarchical model that organizes outline and paragraph data into a directed and ordered tree structure, as shown in Figure 2.

The report template consists of four parts: cover, table of contents, chapter, and back cover. Each chapter can contain multiple sections, each section containing multiple paragraphs (Figure 2a). Paragraphs have four forms: text paragraphs, formulas, graphs, and tables (Figure 2b). One paragraph corresponds to one template, and multiple paragraphs of the same type can also correspond to one template. Different report uses have different formatting requirements, such as differences in the text format of the outlines, paragraphs, and pages (Figure 2c).

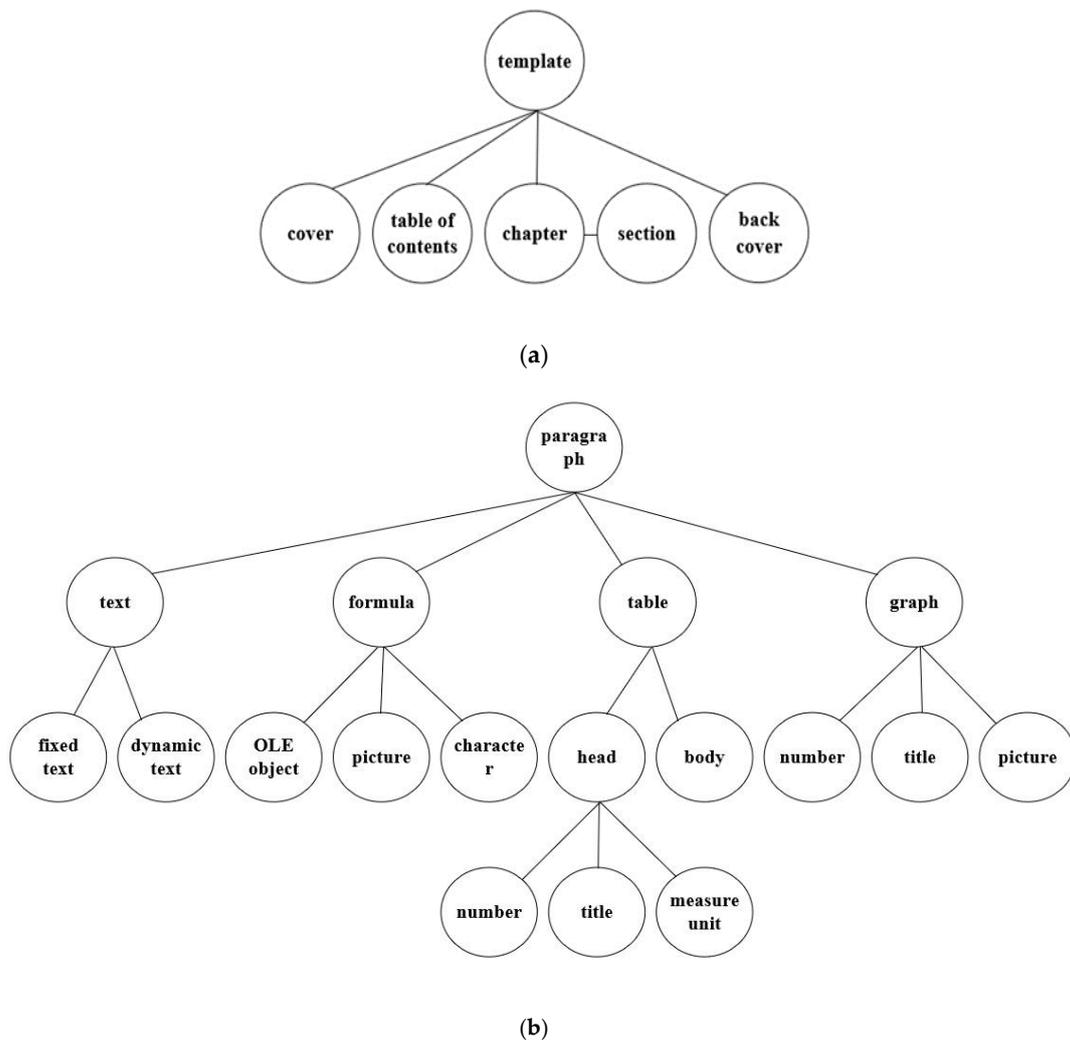
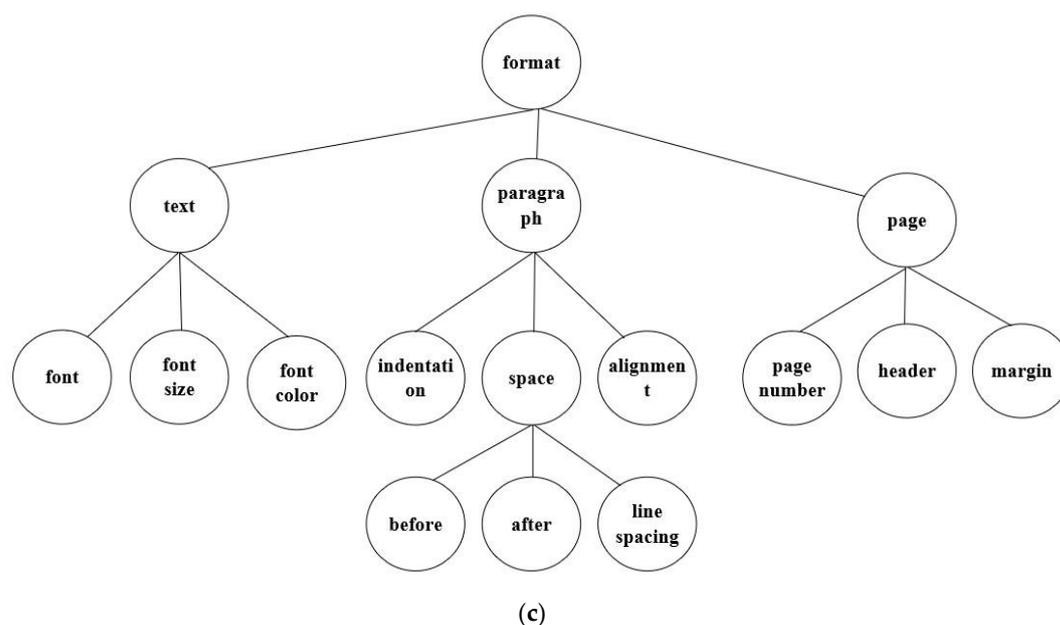


Figure 2. Cont.



**Figure 2.** Hierarchical relations of four-dimensional dynamic template data: (a) outline, (b) paragraphs, (c) formatting.

It is more convenient and flexible to use a graph database to store template hierarchical trees. There are two main types of files to record data in the graph database, namely node and relational storage files. The node storage file stores the node records, the relational storage file stores the relationship records, and the records in each type of file have a fixed length, or number of bytes. When querying data, it is only necessary to calculate the storage location of the record according to the node number, and the data access can be directly performed. The node record contains attributes such as node name, node type, text, parent node name, and number of child nodes, and the relationship record contains attributes that give information about the template relationship.

## 4. Methods and Technology

### 4.1. Related Theory

#### 4.1.1. Relational Database and Graph Database

Storing the template tree structure in a relational database not only produces complicated template tables, but also causes a large number of table connections, which destroys the macro structure of report templates [27]. The graph database is a new database system that efficiently processes intricate relational networks based on the ideas and algorithms of graph theory. In graph theory, the basic elements of the graph are nodes and edges, corresponding to nodes and relationships in the graph database [28,29]. When creating a template, the smallest unit of the report subdivision can be saved as a node, the unit content and the unit attribute are assigned to the node attribute, and the entire report template is finally stored as a large node tree.

#### 4.1.2. Stored Procedure

A stored procedure is a specific collection of T-SQL (Transact Structured Query Language, a database query and programming language for accessing data and querying, update and managing relational database systems) statements executed on the database server that can receive parameters and return status values and parameter values [30]. It is processed as a unit and identified by a name, and the user can call them and return the desired result by giving the name of the stored procedure and the required parameters. When generating an instance of a report, the stored procedure retrieves

template structure data, survey data, and format data based on incoming template information and integrates them into a string for subsequent report generation.

#### 4.1.3. LaTeX Coding

LaTeX coding can fully meet the format requirements of engineering survey reports. Most reports prepared with LaTeX coding have strong logical structure, and need to be divided into chapters, including formula number, illustration number, or cross-reference. When generating the report, the stored procedure converts the retrieved format data into the LaTeX format and writes it along with the content into a tex file, which is easily converted to a PDF report in the LaTeX environment.

#### 4.2. Technical Route

This paper proposes a scheme based on a template tree to generate engineering geological survey reports. The scheme splits the standardized report into multiple parts to construct a hierarchical tree representing the report structure, then stores the tree in the graph database, and finally generates reports dynamically by retrieving the template relational tree and obtaining relevant data. The technical flow diagram is shown in Figure 3.

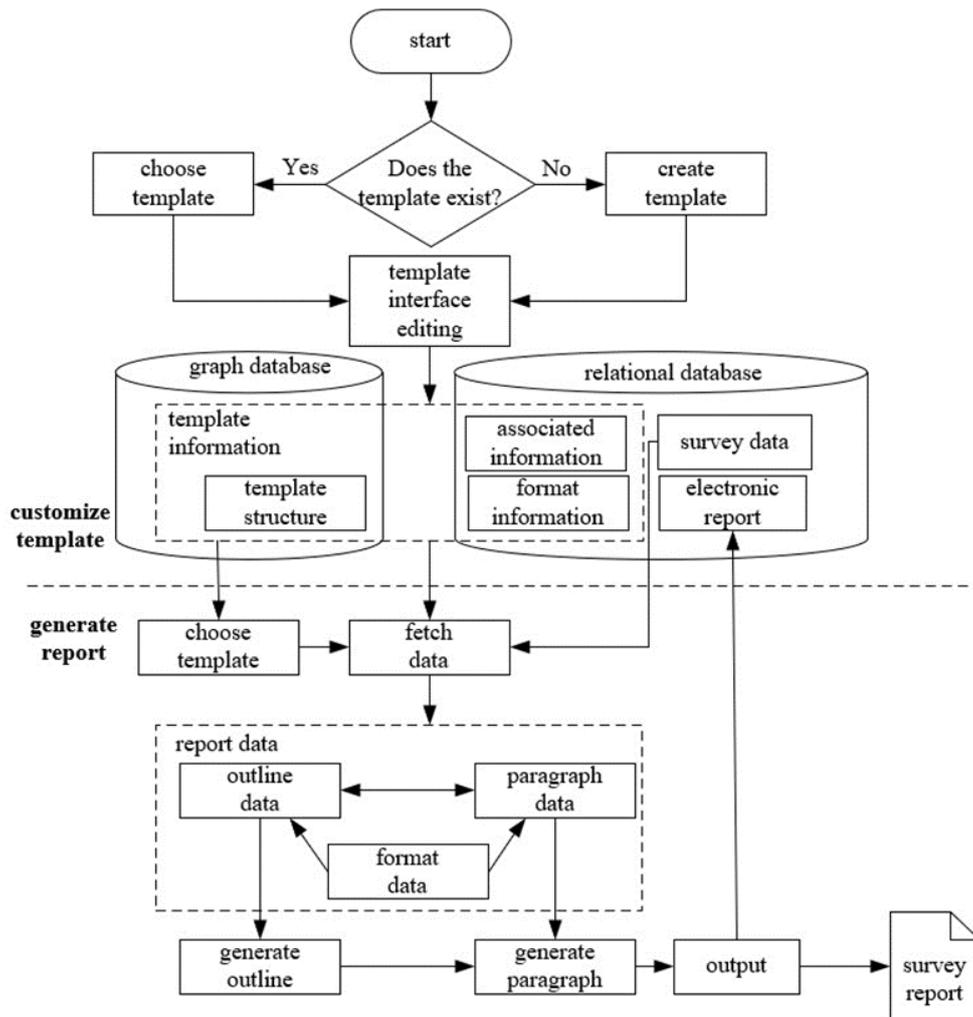


Figure 3. Technical flow diagram.

#### 4.2.1. Template Definition

In the light of the report rules, the standardized report is separated into different chapters, titles and paragraphs. When the paragraph content is text, the text is divided into different sentences. If the text in the sentence contains variable fields, the sentence is divided into fixed and dynamic parts, then the subdivided content is stored as a node, and finally the whole paragraph is saved as a small node tree. When the paragraph content is a graph, different nodes are used to store the information such as captions, numbers, and pictures. When the paragraph content is a table, a cell is saved as a node and the cell information is stored as a node attribute. Finally, the entire report template is stored in the graph database in the form of a large node tree for subsequent management.

#### 4.2.2. Template Editing

If the template is not defined or the template provided by the system cannot satisfy our demand, we must not only input the node tree of the template through the template interface and assign attribute information to each point, but also perform repeated query, modification, and deletion operations to satisfy our dynamic demand. We can also specify the data source for the dynamic node, set and modify template outline titles, paragraph, and page format information such as fonts, line spacing, and indentation in the template interface to satisfy different format requirements for generating the report.

#### 4.2.3. Template Storage

In order to more conveniently realize the data binding of dynamic nodes and report generation, the query result by traversing the template tree of the graph database is stored into the relational database. We input the storage path of the data through the template interface into the relational database for the dynamic nodes, such as the table name, field name, and identification field name. The stored procedure directly obtains the text of the fixed node by traversing the template tree, and indirectly gets the dynamic text by querying the storage path to generate a complete text instance.

#### 4.2.4. Report Generation

When generating a report instance, we first select the template and project, then the stored procedure fetches the template structure data, the survey data, and the format data in the light of the incoming template information. These three types of data are then integrated with LaTeX coding to generate a string containing the template content and format information. Finally, the string is written into a LaTeX file and converted to a PDF report, the report is the final result output by the system. The advantages of this design are that template usage is more intricate, multiple projects can generate reports using the same template, and one project can generate reports using different templates. The part T-SQL procedure that generates an instance of the report is as follows:

```
BEGIN
EXEC procSearchTemplate @reportId, @pageId output, @chapterIds output, @sectionIds output
// Retrieve page formats, outline IDs and paragraph IDs based on template information
EXEC procSearchChapterData @chapterIds, @chapterTexts output
//Retrieve data for each outline
EXEC procSearchSectionData @sectionIds, @sectionTexts output
//Retrieve data for each paragraph
EXEC procPageFormatting @pageId, @pageFTexts output
//Retrieve data for page formats
```

```

EXEC procChapterFormatting @chapterId, @chapterTexts, @chapterFTexts output
// Integrate each outline format and text data

EXEC procSectionFormatting @sectionId, @sectionTexts, @sectionFTexts output
// Integrate each paragraph format and text data

EXEC procIntegrateTemplate @reportId, @pageFTexts, @chapterFTexts, @sectionFTexts,
@templateText output

// Integrate outline data, paragraph data, and page data to generate a string containing template
content and format information

END

```

### 4.3. System Development

#### 4.3.1. Development and Operating Environment

The system utilizes WinForms technology to develop a client/server mode application program and employs SQL Server 2012 and Neo4j 3.1.0 database to store survey, format, and node tree data. The external running environment of the system needs to have Windows Vista or higher operating system with .NET Framework 4.5 or above.

#### 4.3.2. System Architecture

As shown in Figure 4, the system consists of the following six functional modules:

1. Customization management: through the template definition interface (Figure 5), we convert the established Word report template into the document template tree in the system memory, give each node the properties of name, type and text, and establish the relationship between the dynamic part of the template tree and its data source.
2. Interactive management: query, modify, and delete node information to meet template requirements.
3. Retrieval management: a template consists of multiple outline trees and paragraph trees. The correspondence between the template number and the tree is stored in the relation table. We change the template by querying and modifying relation table.
4. Storage management: the storage methods of different forms of templates are different. For example, document templates are saved as files, tree structure templates are saved in a graph database, and template relation records are in a relational database.
5. Data management: the input data of the current system is the result of analysis of the mapping, exploration, and laboratory test data, which is stored in a relational database; the query result data by traversing the template tree of the graph database are stored into the relational database; The stored procedure retrieves template structure data, survey data, and format data based on incoming template information, and integrates them into a string for subsequent report generation.
6. Format management: the report format information can be adjusted according to the report usage or purpose. A template not only contains a variety of format information, but also achieves dynamic text field formatting.

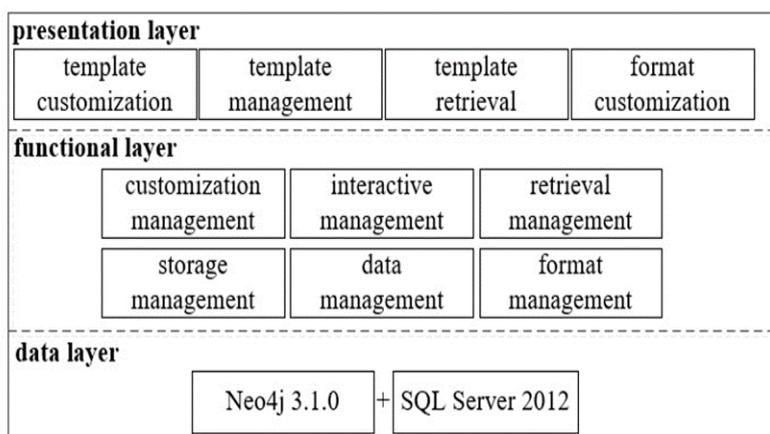


Figure 4. System architecture diagram.

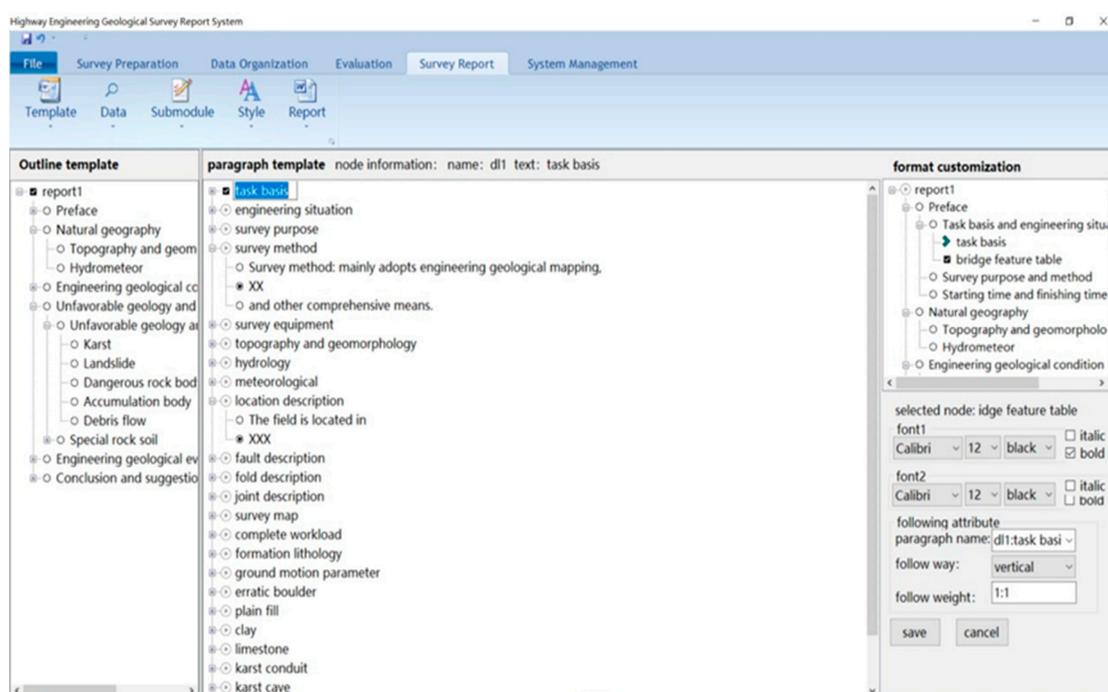


Figure 5. Template customization interface.

## 5. Experiment

### 5.1. Data Description

In the experiment, we used the engineering geological survey data at the construction drawing design stage of Puan No. 1 Bridge in the CZSJ-1 bid section of the Chongqing-Zunyi section (Guizhou border) expansion project of the Lanzhou–Haikou National Expressway, including meteorological and hydrological data, geological structure, and site stability evaluation data. This project is mainly based on surveying and mapping, taking into account tasks such as geological exploration, in situ testing, and indoor testing. We use topographic maps, drilling maps, existing geological data, and roadbed route maps to carry out field exploration and obtain survey data (Figure 6), then comprehensively analyze the mapping, exploration and laboratory test data, and finally and store them in a relational database for report generation.

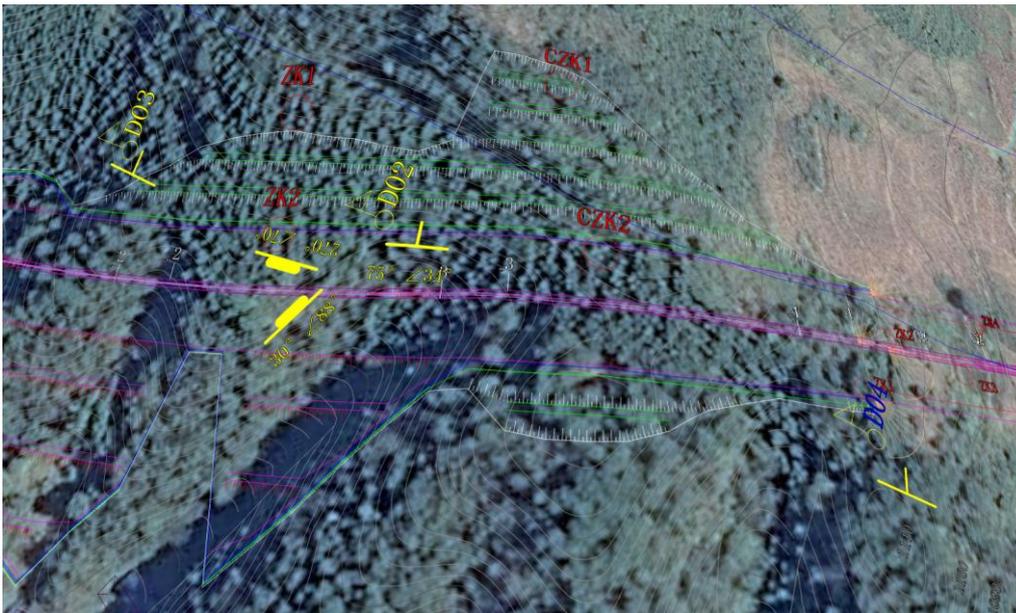
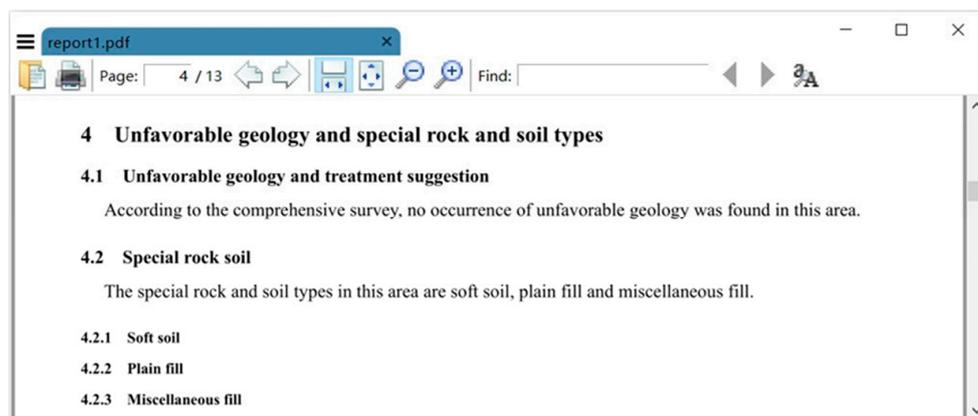


Figure 6. Geological mapping data.

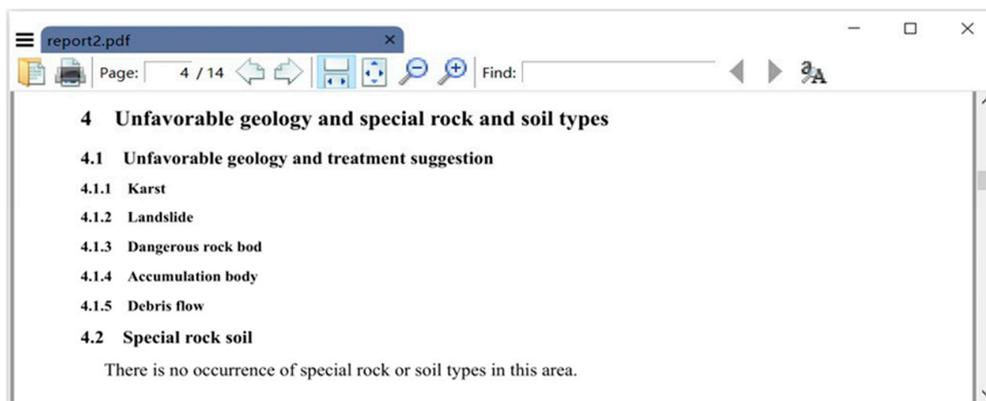
## 5.2. Experimental Results

The system can easily meet the demands for preparation of engineering survey reports. The defined template has strong versatility and flexibility so that the outline structures, paragraph content, dynamic data sources, and format details of the report can be adjusted according to different data. Project B contains data on unfavorable geology and does not contain specific rock soil data. Project A and Project B have the opposite data situation. As shown in Figure 7, the outline structure of the report generated by the two different projects using the same template is fairly different. In Figure 7a, Section 4.1 is no longer divided into sections, and Section 4.2 is divided into three subsections. In Figure 7b, Section 4.1 is further divided into five subsections, and Section 4.2 is no longer subdivided.



(a)

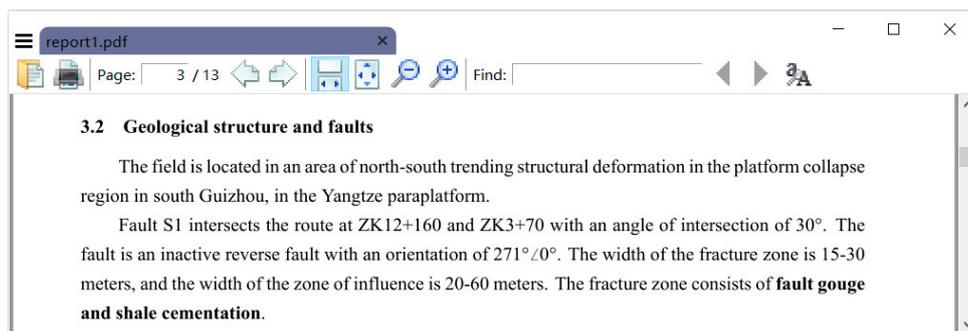
Figure 7. Cont.



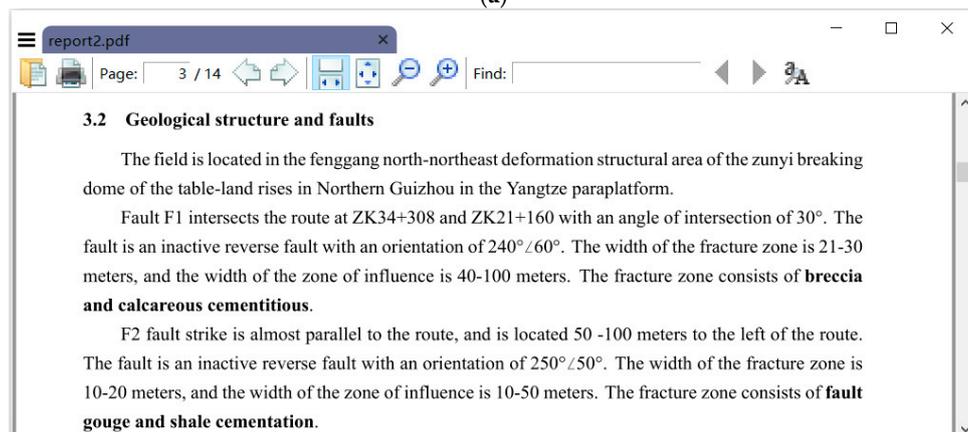
(b)

**Figure 7.** Comparison of outline results for two different projects using the same template to generate reports: (a) Project A and (b) Project B.

The fault data in Project A and Project B is different. As shown in Figure 8, the paragraph text of the two reports generated using the same paragraph text template is quite different. In Figure 8a, only one fault segment is included under Section 3.2. In Figure 8b, Section 3.2 contains two fault segments, which are generated from the same fault segment template.



(a)



(b)

**Figure 8.** Comparison of paragraph text results for two different projects using the same template to generate reports: (a) Project A and (b) Project B.

Project A and Project B have different field joint characteristics data. As shown in Figure 9, the number of rows in which the two different items use the same template to generate the table differs depending on the number of field joints included in the two items.

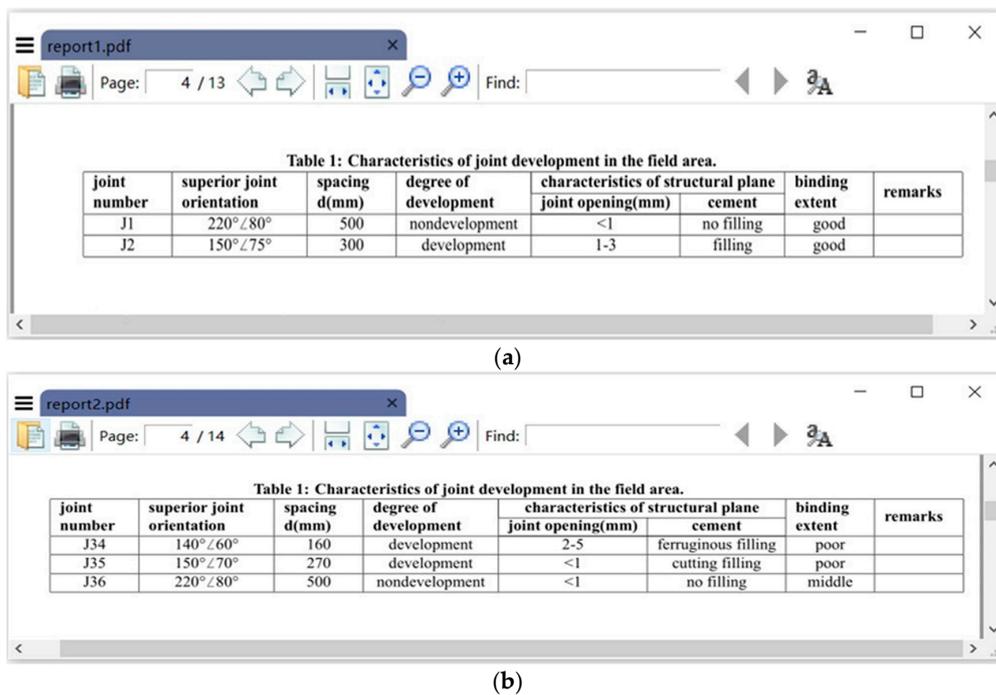


Figure 9. Comparison of tables generated for two different projects using the same template to generate reports: (a) Project A and (b) Project B.

Template A and template B differ in the outline structure, intra-segment structure, and format details. As shown in Figure 10, the outline structure of the report generated by the same project data using the two different templates is different. In Figure 10a, Section 1.1 is divided into two subsections. In Figure 10b, Section 1.1 is divided into three subsections.

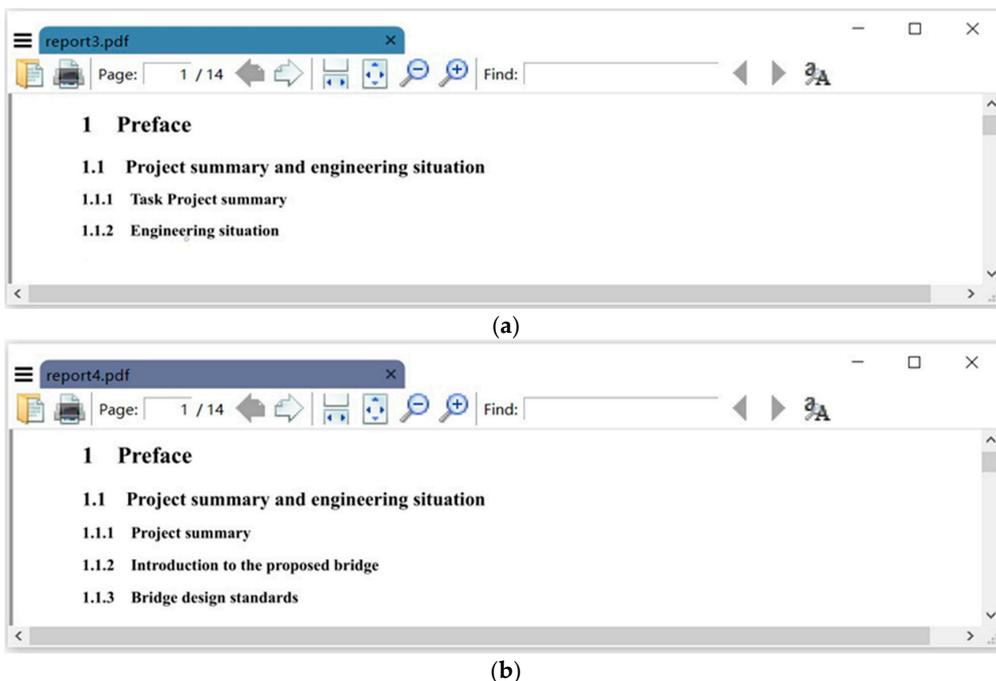
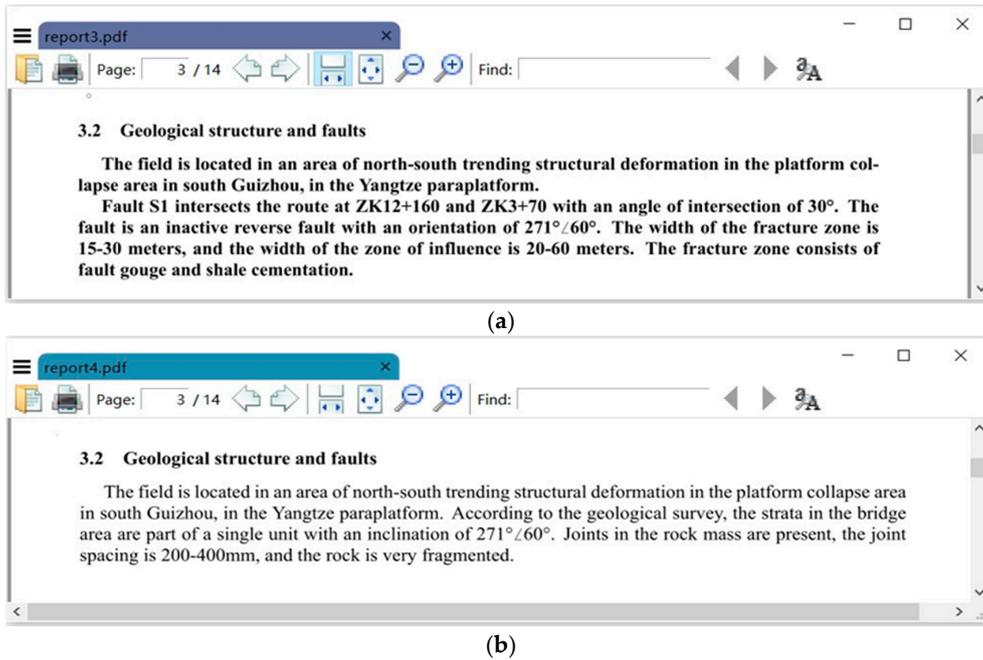


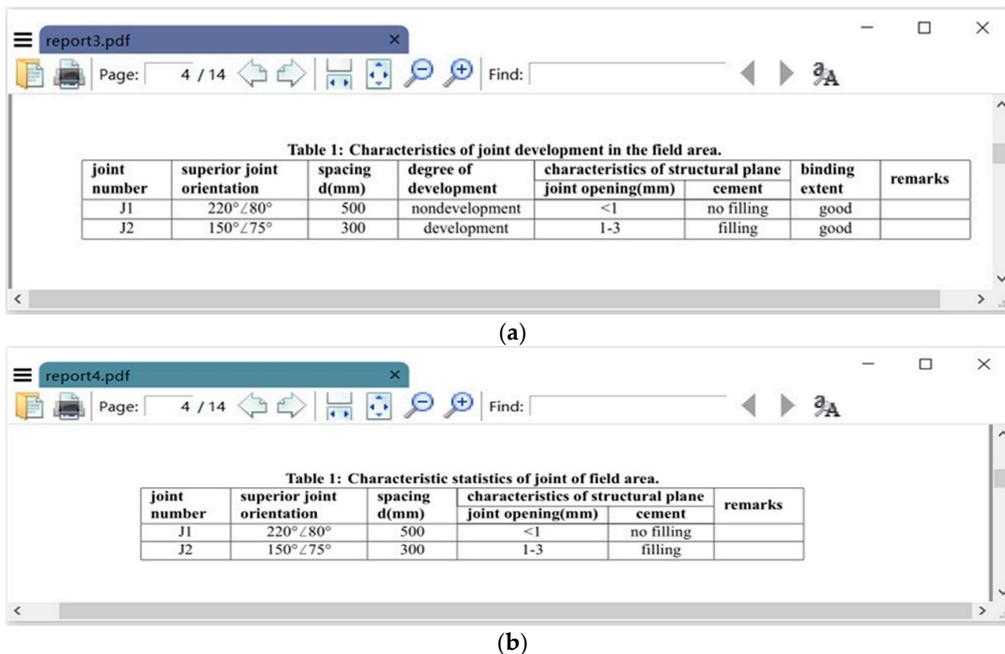
Figure 10. Comparison of outlines generated using two different templates for the same project: (a) Template A and (b) Template B.

As shown in Figure 11a,b, the same section in the two different templates contains different numbers of paragraphs, and the paragraph text for the same project is also different between the two templates.



**Figure 11.** Comparison of paragraph text generated using two different templates for the same project: (a) Template A and (b) Template B.

As shown in Figure 12, the table generated for the same project using the two different templates has a large difference; the table title in Figure 12a is different from that in Figure 12b, and the header font and number of columns also differ.



**Figure 12.** Comparison of tables generated using two different templates for the same project: (a) Template A and (b) Template B.

### 5.3. Effect Analysis

Following the actual data analysis, we will now compare the implementation results of the document generation method based on FDDT with other document generation methods based on template (existing document generation methods based on template that are different from the document generation method based on FDDT) with respect to the following aspects:

- **Outline structures:** in the document generation method based on FDDT, the report outline structure can be adjusted with the presence or absence of dynamic data in the template, that is, the dynamic data greatly affects the outline structure of the final generated report. As shown in Figure 7a,b, the existence of special rock soil data determines whether Section 4.2 is further sub-sectioned. Once the outline structure of the template used by other document generation methods based on template is determined, it is difficult to modify, and the outline structure will not be adjusted according to the dynamic data, which have low flexibility.
- **Paragraph content:** when there is a multi-branch sentence in a text paragraph, the document generation method based on FDDT can choose different branches according to varied conditions in order to better adapt to the changing characteristics of that paragraph. As shown in Figure 8b, if the intersection of two faults and the route is different, the selected fault description sentence is also different. Other document generation methods based on template can only generate reports with fixed paragraph structures, and do not involve automatic selection of sentences within the paragraph. The document generation method based on FDDT can also generate tables with complex styles and flexible growth of rows and columns. By comparison, the structure and format of the table template used in other document generation methods based on template are simple and fixed. These are not suitable for generating complex dynamic tables.
- **Format details:** in the document generation method based on FDDT, the report format information can be adjusted in the light of the report usage or purpose. A template not only contains a variety of format information, but also achieves dynamic text field formatting. As shown in Figure 8a, the composition of the fracture zone is displayed in bold. While the template format used by other document generation methods based on template is fixed, this approach can achieve single fixed text formatting, but does not involve dynamic text field formatting. For example, paper [23] requires that the format of the table and the content of the header should be fixed according to the standard format of the report when creating a template.
- **Dynamic data sources:** In the document generation method based on FDDT, there are various methods for obtaining dynamic data, such as SQL Query, stored procedures, and functions. As shown in Figures 8a and 9a, the composition of the fracture zone is obtained by SQL Query, and the table content is the return value of stored procedure. These methods are similar to some document generation methods based on templates, such as the data query method in paper [23] and the data processing method in paper [24]. The primary difference is that the data source methods of the document generation method based on FDDT can be dynamically adjusted, different source methods can be selected flexibly in the light of the needs when defining different templates, while the data source methods of other document generation methods based on template are fixed. For example, paper [23] and paper [24] use SQL statement to query data.

When multiple templates have been entered in the system, the document generation method based on FDDT only needs to change or improve the template record table, and does not need to change the program code, so as to achieve efficient assembly of the template to meet the generation requirements. However, other document generation methods based on templates cannot meet all-round dynamic changes in four aspects: outline structure, paragraph text, format details, and data source. The natural language generation methods are incapable of generating normalized documents in such professional fields.

## 6. Conclusions

In this paper we considered the problems in the preparation and management of engineering survey reports and analyzed the multidimensional variable features of professional field documents. Our solution proposes a model of standardized reports based on a four-dimensional dynamic template, and we implemented the model by designing and developing a generation system for survey reports. Results indicate that the model can solve the problems of chapters, paragraphs, and format details being flexibly changed and adjusted according to dynamic data changes, and the report generation system has the ability to satisfy the requirements of the preparation and management of engineering survey reports and also improves the work efficiency and data accuracy.

Several future working directions are suggested in this research field. First, the format adaptive technology of node text should be studied, considering the format settings of different types of text such as the display format of numbers and text in the same position. Secondly, the related changes in the content of cross-chapter paragraphs should be studied. We should consider the possible relationships (quantitative relationship, causality) between tree nodes, such as the content of a paragraph affects the generation of subsequent chapters and paragraphs in addition to the study of the change characteristics of report based on tree structure. Thirdly, the combination of the document generation method based on FDDT and artificial intelligences should be studied, considering mining and extracting templates from a large number of existing reports to achieve the intelligent generation of standardized reports.

**Author Contributions:** Conceptualization, Jiqui Deng, Yujiao Lei and Jian Lin; Methodology, Jiqui Deng and Yujiao Lei; Software, Yujiao Lei; Investigation, Jian Lin and Chaoyue Liu; Writing—Original Draft Preparation, Yujiao Lei; Visualization, Yujiao Lei; Writing—Review and Editing, Jeffrey M. Dick and Mohammad Naser Lessani. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Key R&D Program of China (Project No. 2017YFC0601503, No.2018YFC0603902) and National Natural Science Foundation of China (Project No. 41401532).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Liang, H.K. Discussion on the preparation of highway engineering geological survey report. *Commun. Sci. Technol. Heilongjiang* **2015**, *38*, 10. [[CrossRef](#)]
2. Gao, X.Z. Analysis and summary of engineering examples of misuse of normative formulas in engineering geological survey reports. *Sci. Technol. Inf.* **2013**, *25*, 325–326.
3. Ma, Y.Z.; Zhang, J.P.; Zheng, Y.H. Research on Automatic Generation of Documents for Automotive Radiator Design Calculation Platform. *J. Qingdao Univ. Eng. Technol.* **2013**, *28*, 71–75.
4. Jiang, P.; Xu, F.; Qi, R.Z. Construction of a Flood Control Document Intelligent Generation Model Based on Cloud Platform. *Water Resour. Informatiz.* **2013**, 25–32.
5. Chen, Y.Y.; Xue, J.N.; Yu, H.H. Quick and automatic generation method for the evaluation report of the small hydropower substitute fuel project. In Proceedings of the 9th IFIP WG 5.14 International Conference on Computer and Computing Technologies in Agriculture, Beijing, China, 27–30 September 2015; pp. 64–71.
6. Reiter, E.; Mellish, C.; Levine, J. Automatic generation of technical documentation. *Appl. Artif. Intell.* **1995**, *9*. [[CrossRef](#)]
7. PadmaLahari, E.; Kumar, D.S.; Prasad, S. Automatic text summarization with statistical and linguistic features using successive thresholds. In Proceedings of the 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Syed Ammal Engineering College, Ramanathapuram, India, 8–10 May 2014; pp. 492–496.
8. Khan, A.; Salim, N.; Farman, H. Abstractive text summarization based on improved semantic graph approach. *Int. J. Parallel Program.* **2018**, *46*, 992–1016. [[CrossRef](#)]
9. Kupiec, J.; Pedersen, J.; Chen, F. A trainable document summarizer. In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, 9–13 July 1995.

10. Serban, I.V.; Lowe, R.; Charlin, L. Generative deep neural networks for dialogue: A short review. *arXiv* **2016**, arXiv:1611.06216.
11. Clerwall, C. Enter the robot journalist: Users' perceptions of automated content. *Journal. Pract.* **2014**, *8*, 519–531. [[CrossRef](#)]
12. Abdul-Kader, S.A.; Woods, J.C. Survey on chatbot design techniques in speech conversation systems. *Int. J. Adv. Comput. Sci. Appl.* **2015**, *6*, 72–80.
13. Klisarova-Belcheva, S.; Ilieva, G.; Yankova, T. Business intelligence and analytics—contemporary system model. *Trakia J. Sci.* **2017**, *15*. [[CrossRef](#)]
14. Li, H.; Chiang, H.D.; Gale, W.G. A novel system for automatic generation of service reliability reports from automated mapping and facility management (AM/FM) systems. *IEEE Trans. Power Syst.* **2016**, *17*, 812–817. [[CrossRef](#)]
15. Colineau, N.; Paris, C.; Linden, K.V. Automatically generating citizen-focused brochures for public administration. In Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times, College Park, MD, USA, 12–15 June 2011; pp. 10–19.
16. Li, Q.; Liu, R.K.; Sun, Q.X. Research of automatic progress report generation for railway construction projects in China. *Adv. Mech. Eng.* **2015**, *7*. [[CrossRef](#)]
17. Jia, Q.Z.; Xiao, B. The universalization of data interaction technology research between database and spreadsheet of WORD based on OLE. In Proceedings of the 2016 International Conference on Sustainable Energy, Environment and Information Engineering, Bangkok, Thailand, 20–21 March 2016; pp. 181–186.
18. Yang, H.J.; Jia, Q.Z. Automatic synchronization technology of report data based on OLE. In Proceedings of the 2nd International Conference on Materials Science, Resource and Environmental Engineering (MSREE 2017), Wuhan, China, 27–29 October 2017.
19. Huynh, E.; Roach, J.D.; McLaren, T. A computer-assisted method for pathogenicity assessment and genetic reporting of variants stored in the Australian Inherited Retinal Disease Register. *Australas. Coll. Phys. Sci. Eng. Med.* **2016**, *39*, 239–245. [[CrossRef](#)] [[PubMed](#)]
20. Lu, B.Y.; Yang, X.F. Generating Word report and dynamic structure table with Delphi. *Comput. Appl. Softw.* **2007**, 180–183. [[CrossRef](#)]
21. Ge, F.; Wu, N. Platform for automatically producing word document based on multiple techniques. *J. Univ. Electron. Sci. Technol. China* **2007**, *36*, 263–266.
22. Cai, X.; Shen, W.; Li, L.; Ge, Y.; Lu, L.; Zeng, L. Research and Implementation of Word Documents Generation Based on C#. In *2015 International Conference on Electronic Science and Automation Control*; Atlantis Press: Amsterdam, The Netherlands, 2015. [[CrossRef](#)]
23. Dong, G.Z. Design and Implementation of Logging Report Generation System Based on .NET. *Mud Logging Eng.* **2014**, *25*, 70–72.
24. Liu, P.; Qin, W.; Zhou, Y.H. Automatic generation method of word report based on dynamic web page. *Comput. Eng.* **2012**, *38*, 279–284. [[CrossRef](#)]
25. Qu, M.C.; Liao, M.H.; Wu, X.H. Construction and application of an automatic document generation model. *Comput. Integr. Manuf. Syst.* **2008**, 1297–1305.
26. Jiang, C.M.; Wu, X.H.; Qu, M.C. Automatic document generation strategy for graphic-oriented modeling and automatic code generation platform. *J. Theor. Appl. Inf. Technol.* **2012**, *45*, 342–349.
27. Robinson, I.; Webber, J.; Eifren, E. Options for storing connected data. In *Graph Databases*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2013; pp. 11–23.
28. Miller, J.J. Graph database applications and concepts with Neo4j. In Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA, 23–24 March 2013; pp. 141–147.
29. Van Bruggen, R. Graphs and graph theory—an introduction. In *Learning Neo4j*; Packt Publishing Ltd.: Birmingham, UK, 2014; pp. 7–20.
30. Sunderic, D. *Microsoft SQL Server 2005 Stored Procedure Programming in T-SQL & .NET*; McGraw-Hill, Inc.: New York, NY, USA, 2006.

