List of Tables

Table of Contents

The problem statement was also passed as input to the class diagram generator and the following timings were recorded:

Analysis Time (AT): 12.58s

Diagram Generation Time (DGT): 10.50s

Overall time (OT): 23.08s

It is to be noted that the recorded time is taken to be the average of worst case scenario, which is when when the tool is ran for the first time.

…….

6. The disparity seen between the outputs generated by the testers and the tool is assumed to be the subjective assessment of the problem statements by the testers unlike the objective and rule based assessment done by the tool.

Table 6.9: Inferences table

Characteristic	**Traditional Method**	**Class Diagram Generator Method**	**Inferences**
Accuracy - Classes	60.92 (%)	90	The tool shows higher accuracy in identifying classes.
Accuracy - Attributes	55.42	88.3	The tool shows higher accuracy in identifying attributes classes.
Accuracy - Methods	38.36	100	The tool shows higher accuracy in identifying methods of classes.
Sensitivity - Classes	68 (%)	90	The tool has higher sensitivity when identifying classes.
Sensitivity - Attributes	44	94.32	The tool has a higher sensitivity when identifying attributes.
Sensitivity - Methods	45.33	100	The tool has higher sensitivity when identifying methods.
Diagram Analysis Time	112.4 (seconds)	12.58 (seconds)	The tool analyzes input statements faster than an average tester.
Diagram Generation Time	278.5	10.50	The tool generates output faster than an average tester.
Overall Time	390.9	23.08	The overall time required by this tool is far smaller than what is required by an average tester.

Table 6.10: Comparison Table between similar tools

Support	CM-Builder	LIDA	GOOAL	NLOOML	DC-Builder	RACE	RAUE	RAPID	ACDG
Classes	Yes	User	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Attributes	Yes	User	Yes	Yes	Yes	Yes	Yes	No	Yes
Methods	No	User	Yes	Yes	No	Yes	No	No	Yes
Associations	Yes	User	Semi-NL	No	Yes	Yes	Yes	Yes	Yes
Multiplicity	Yes	User	No	No	No	No	Yes	No	No
Aggregation	No	No	No	No	Yes	Yes	Yes	Yes	No
Generalization	No	No	No	No	Yes	Yes	Yes	Yes	No
Instances	No	No	No	No	No	No	No	No	No

CHAPTER 8
CONCLUSION

The research to compare the effectiveness of the proposed class diagram generator was performed by comparing the class diagrams generated by the proposed system to the diagrams generated by 10 final year computer science students and has been found that the diagrams generated by the proposed system were more accurate and the diagrams were generated faster. The research can also be further extended to find the optimum level of automation, such that the proposed system and other such systems can offer the right balance between automation and freedom to the software engineers. In addition, the research can also be extended to find the cost to companies which implement the proposed system.

The tool also supports basic relationships among classes showing is-a or has-a relationships. Support for more complex relationships haven't been fully developed for the tool and the research can be extended in this regard. The tool can be further improved by adding features that allow the users to edit the existing diagram as well introduce new elements into the diagram. This will allow the tool to vastly contribute to the Design phase if the Software Development Life Cycle in the long run.

Therefore, with the proposed class diagram generator, the class diagrams were generated from the input natural language text and have shown to be much faster and accurate than the traditional method of class diagram generation, which can extensively conclude that artificial intelligence can make the design phase of software development a lot more efficient for software developers.