

Evaluation Summary

User Evaluation Summary

Process

At the beginning, 5 participants are invited to join the first evaluation and complete the questionnaires. However, I have decided to fix the bugs between the time of evaluations so that the version used by different participants is different. This decision has added an extra variable to the software which is not a good approach.

After discussion with my supervisor, I have decided to use a fixed version of software (archived in git branch *Test/UserEvaluation1*). This time, 10 participants are invited to join the evaluation and complete the questionnaires. However, the face-to-face evaluation is forced to be cancelled due to force majeure. The result gathered from these 10 participants will be documented below.

Questions

The questionnaire asks the participant to briefly design an architecture for a video game.

Question 1 asks about the types of game and participants want to design, all participants want to design video games using client-server or distributed system architecture styles; Question 2 asks participants to act as architects and design an architecture model for project managers. This question potentially asks participants to design the logical view for their games. 9 participants build client and server components at different levels and 1 participant uses the distributed approach.

Question 3 asks participants to act as architects and design an architecture model for a software engineer. This question potentially asks participants to design the development view or deployment view for their games. 8 participants use the object-oriented approach to design a UML diagram with classes, attributes and methods (development view), while 2 participants particularly model the hardware deployments with database storage, distributed servers and client hardware (deployment view).

Multiple choice feedback

Question 4 and 5 asks for the user experience and feedback/suggestions from the participants:

- Question 4a asks the participants if they can find out any corresponding components, interfaces and connectors between their views. This question evaluates the consistency of participants' input and their feelings. 7 participants can find 1 corresponding entity, and 3 participants can find 2 corresponding entities.
- Question 4b asks the participants if they have used the design experience from their first views in designing the second one. This question evaluates if participants can feel the connections between their models. All participants answer 'Yes' to this question. In particular, some participants state that their second views are based on the decomposition of their first views.
- Question 4c asks the participants if they can identify any inconsistencies between their views. The evaluation is performed before the constraint checkers are

implemented so this question lets participants self-evaluate their views. All participants answer 'No' to this question.

- Question 5 evaluates the user experience from the feedback and suggestions. Each question uses a scale from positive attitude to negative attitude. The number in the table below indicates the number of participants.

	1 (Very negative)	2 (Negative)	3 (Moderate)	4 (Positive)	5 (Very positive)
The visualization is expressive and effective	0	1	1	6	2
The user control is easy-to-use and flexible	(Bad) 0	/	(Moderate) 3	(Good) 5	(Very good) 2
Not enough functionalities are provided	(Always) 3	(Sometimes) 5	/	(Rarely) 1	(Never) 1
The tool is clearly structured	0	1	2	3	4

State-of-the-art Evaluation

Background

This is a self-evaluation for ArchPrime. It compares ArchPrime's functionalities and non-functional performances with some state-of-the-art modelling tools. The first tool is draw.io (<https://www.diagrams.net/>), which is a general UML diagram maker. The second tool is Structurizr (<https://structurizr.com/>), which is an online architecture development environment based on the C4 model. The third tool is ArchStudio (<http://isr.uci.edu/projects/archstudio/>), which is another architecture development environment based on the Eclipse IDE.

Functionality / Non-functional property	ArchPrime performance	Draw.io performance	Structurizr performance	ArchStudio performance	Personal ranking for these tools
Visualisation					
Support textual or graphical visualisation on architecture description	ArchPrime supports both xADLite-based JSON data representation and lines-and-boxes visualisation	Draw.io supports free textual or graphical visualisation, it does not have inherent architecture description modelling functions.	Structurizr supports both code-based and diagram-based visualisation on the architecture description at 4 levels of details.	ArchStudio supports a special ArchStudio Architecture Description format that can be used for both textual (ArchEdit) and graphical (Archipelago) visualisations.	1. Structurizr 2. ArchPrime & ArchStudio 3. Draw.io
Support rich graphical visualisation elements for architectural entities	ArchPrime only supports lines, 2D shapes and simple text formats.	Draw.io supports abundant graphical visualisation elements that can be used to design different types of diagrams, these elements can also be used when designing architecture diagrams	Structurizr supports a moderate number of elements for different entities. For example, it represents stakeholders as a person-shape component.	ArchStudio supports basic line-and-boxes visualisation with simple colours and formats.	1. Draw.io 2. Structurizr 3. ArchPrime & ArchStudio
Support easy-to-use and informative GUI with reactive	ArchPrime supports simple layout (navigation,	Draw.io is designed with a clear layout and an informative	Structurizr supports two kinds of GUI: code-based GUI	ArchStudio uses the layout based on Eclipse IDE, some functions	1. Draw.io 2. Structurizr 3. ArchPrime 4. ArchStudio

user interactions	topbar, workbench) with graphical visualisation workbench. The performance is moderate according to user evaluation.	workbench. It is sensible with user interactions.	and diagram-based GUI. Both of them have easy-to-use toolbars, informative GUI elements and are reactive to user inputs.	may be ambiguous to users. It illustrates the architecture description using an information-rich UI.	
The tool can be easily downloaded, installed and utilised by architects	ArchPrime is a web application specifically for the browser environment.	Draw.io is a free web application with multi-platform support.	Structurizr is a web service with multi-platform support.	ArchStudio is based on Eclipse IDE. It needs to be downloaded, installed and then used.	1. Draw.io 2. ArchPrime & Structurizr 3. ArchStudio
Architecture Description Representation					
Support architects to build an architecture model from a specific viewpoint	ArchPrime provides a modelling tool based on view and viewpoint approach.	Draw.io does not have inherent architecture modelling supports	Structurizr is able to provide an architecture model using the C4 approach.	ArchStudio provides the functionalities of modelling based on architects' decision	1. Structurizr & ArchStudio & ArchPrime 2. Draw.io
Support architects to build hierarchical architecture model at different levels of details	ArchPrime supports hierarchical structure inside a view.	Draw.io only supports single-level diagrams.	Structurizr divides the model at 4 different levels: context, container, component and code.	ArchStudio uses xADL to manipulate data, it can be used to build hierarchical models.	1. Structurizr & ArchPrime & ArchStudio 2. Draw.io
Support architects to build multiple views for an architecture description	In one architecture description, ArchPrime allows architects to create an arbitrary number of views.	Draw.io is able to store multiple diagrams in one diagram file. The diagrams cannot be inherently linked together.	The C4 model provides a single comprehensive view. It can be used repeatedly for different viewpoints.	Archipelago supports users with multiple views for an architecture description.	1. ArchPrime & ArchStudio 2. Structurizr 3. Draw.io
Support architecture analysis and checking tools	ArchPrime has a tree view menu and a constraint checker.	Draw.io does not have inherent architecture-related functions.	Structurizr supports a lot of architecture analysis tools.	ArchStudio has Archlight, which is a built-in analysis tool	1. ArchStudio & Structurizr 2. ArchPrime 3. Draw.io

Heuristic Evaluation

Background

This is another self-evaluation for ArchPrime. It uses Nielsen's heuristic evaluation approach (Nielsen, J. (1994)) to inspect the usability of ArchPrime. In this evaluation, 10 heuristics are given and the performance of ArchPrime regarding these heuristics are evaluated.

Nielsen's heuristics	Explanation	ArchPrime Usability
Visibility of system status	The system should always keep users informed about what is going on.	1. The navigation drawer and application bar help the users to keep track of their current architecture description file and view. 2. The tree view menu helps the users to identify the structure of the current architecture model.
Match between system and the real world	The system should speak the users' language, with words, phrases and concepts familiar to the user.	1. ArchPrime uses English as the default application language. 2. In the modelling process, ArchPrime uses unanimous architectural terminologies based on IEEE 42010 standard. 3. ArchPrime provides explanations for constraints so that users can easily understand which entity is inconsistent.
User control and freedom	Users often choose system functions by mistake and will need a clearly marked 'emergence exit' to leave the unwanted state.	1. On the workbench, ArchPrime provides 'Back' and 'Enter' functions for users to freely go up/down between the different levels of a model. 2. Components, connectors and interfaces can be easily deleted, repositioned and modified.
Consistency and standards	Users should not have wonder whether different words, situations or actions mean the same thing	1. ArchPrime uses consistent terminologies and semantics.
Error prevention	Either eliminate error-prone conditions or check for them and present users with a confirmation option.	1. Operational errors such as misclicking and mistyping can be easily reverted and fixed. 2. Some architectural errors such as inconsistencies can be figured out by the constraint checker.
Recognition rather than recall	Minimise the user's memory load by making objects, actions, and options visible.	1. ArchPrime allows user actions such as drag-and-drops clicking and text typing through mouse and keyboard. 2. The application bar at the top provides some helpful tools when designing the model.

Flexibility and efficiency of use	The system can cater both inexperienced and experienced users.	<p>1. For inexperienced users, ArchPrime provides the explanation on the context menu (right-clicking menu) and the constraint checker for novice architects to adjust their models.</p> <p>2. For experienced users, ArchPrime provides a flexible definition of components, connectors and interfaces for professional architects to adapt these entities into different views.</p>
Aesthetic and minimalist design	Dialogues should not contain information which is irrelevant or rarely needed.	1. The dialogue from constraint checker displays the necessary information and fixing advice without irrelevant words.
Help user to recognise, diagnose, and recover from errors	Error messages should be expressed in plain language.	1. The dialogue and error message explains why the architectural entities or operations are invalid.
Help and documentation	Provide help and documentation.	<p>1. ArchPrime provides an 'Example' description for new users to study the tool.</p> <p>2. There is a user manual documenting how to use the tool.</p>