# Comparison

Many different adaptive hypermedia engines have been developed over the last two decades. In this section ALAT will be compared to several authoring tools of these engines. These tools have been selected in order to point out interesting comparisons in design philosophy and functionality. This comparison also provides an indication as to the relevance of ALAT in the field of adaptive hypermedia. The comparisons are followed by a discussion on major differences in functionality and design philosophy. Various points made in this discussion will be used to contribute to [CHAPTER REF], which deals with possible future work regarding ALAT.

Many early adaptive hypermedia engines provided only limited support for authoring. Prime examples of this are Interbook, as discussed in section 2 and KBS Hyperbook [QuNe02]. Interbook used MS Word for authoring and KBS Hyperbook relies on the creation of XML files to create adaptive applications. These early forms of authoring do not provide a lot of user support and require the user to provide either annotations or adaptation code manually. More modern systems such as AHA! use extensive user models and have a separation between authoring adaptation and authoring content and presentation. This allows the system to manage nearly arbitrary adaptation rules and arbitrary presentation [Sm11]. The following comparisons have been made on dedicated authoring environments which author domain and adaptation models rather than content and presentation.

Apart from hypermedia engines that support almost arbitrary adaptation rules, there also exist modern systems which do not focus on this notion. An example of such a system is ACTSim [GaDa10]. ACTSim is a “unique composition tool that supports the rapid development of personalized training simulation”. Its focus is making educational soft skill simulations adaptive. The creators of ACTSim claim these simulations are difficult to author as is, yet ACTSIM provides enough support for a non-technical expert to create simulations with ease. When comparing ACTSim to ALAT it becomes apparent that the scope of adaptive hypermedia that can be authored with this tool is fairly limited. This limitation however, is a double edged sword. Because of the limitations in authoring scope, ACTSim can be used to author educational simulations to a much further extent than can be achieved by ALAT. When creating an authoring tool, a tradeoff has to be made between specialization and genericity. As mentioned multiple times earlier in this thesis, ALAT aims to be as generic as possible. The comparison with ACTSim shows another possible design philosophy with regards to specialized adaptive hypermedia authoring tools as opposed to more generic tools

WOTAN [FrRo05] has tools that take an interesting approach in user interface design. It has tools for both an indented list hierarchy as well as an interface representing the current project in a directed graph. This, in essence, is a strong set of views. The downside of this representation is again the graph implementation. Though some visualization techniques such as automating the layout and clustering groups of nodes are added to prevent clustering, larger projects still suffer from this graph representation in terms of complication. On top of this, these visualization techniques make the graph a lot more complicated. Another feature of WOTAN that brushes against the principles of ALAT is the representation of adaptation rules. These rules are represented as items in the project hierarchy or as nodes in the graph representation. This causes a conflict of interest in the adaptive course design as both the domain as well as the adaptation model are overlapping in the main project structure interface. The interface also becomes more confusing when concepts which have multiple parent concepts are introduced. The result of this is that these concepts appear multiple times in the indented list project hierarchy. This is very confusing, which is why ALAT does not allow these kind of domain connections (and uses a tree structure instead).

Whereas WOTAN had some issues regarding conflicts of interest, MOT[[1]](#footnote-1) 3.0 [FoCr10] focusses intensely on consistency regarding a ‘separation of concerns’. It does this by closely following the five-layer LAOS framework [CrMo03]. The corresponding layers are: Domain model, Goal model, User model, Adaptation model and Presentation model. MOT is a web-based authoring system used for on-line adaptive course production. It is used in combination with the PEAL adaptation strategy author to author adaptive hypermedia applications. It relies on other adaptive hypermedia engines such as AHA! to deliver its courses [CrSm05].

Because of the extensive layering and separation of concerns in MOT, its authoring process is more modular as opposed to the one-stop-shop process in ALAT. Domain models are constructed in a dedicated web-interface and the adaptation model is created using the PEAL adaptation strategy author. This is an application which supports the user in writing the adaptation code by implementing status bar suggestions to improve code validity, code completion as well as a strategy wizard to define and initialize variables. Apart from this separation, attributes that should be stored in the user model are declared separately from the ones that shouldn’t (the former are declared in the web interface, the latter in PEAL).

The big differences between MOT and ALAT can be explained when comparing their respective design philosophies. MOT is created such that content authors can set up a domain using the MOT web interface and adaptation authors use PEAL to create the appropriate adaptation strategy. ALAT uses the role of an adaptation author to set up templates, such that the content author can create an entire adaptive application by himself.

The use of multiple tools and the lack of templating does make authoring in MOT more complicated. Even though PEAL provides support when creating the adaptation model, all standard behavior still has to be created separately for every concept. The use of different tools also creates a problem which is present in GAT as well: When an author wants to change the domain whilst creating the adaptation model, he then has to move back and forth between tools to achieve this.

Another more recent authoring environment to take a look at is AMAS [HaCo11]. This is an adaptive educational hypermedia project in which user experience is the main focus. It is designed to be usable by teachers. This is done by providing by implementing reusable assets in terms of both content as well as adaptation strategies. Integrated group adaptation is implemented to help groups of students that are struggling with a particular topic. This kind of adaptation, while limiting the level of concept customization, greatly decreases the complexity of authoring an application. Special attention is paid to User experience (UX) as well. Gaffney, Conlan and Wade [GaCo14] claim that “*Inadequate UX design of AEH authoring tools may be a key factor impeding their widespread commercial and academic uptake*”. This was measured by evaluating Style, Color, Look & Feel and Familiarity through questionnaires. Even though it might be difficult to measure and evaluate this quantitatively (apart from querying user opinions through surveys) it is clear that an effort has been made to prevent a “nerdy” interface.

When comparing AMAS to ALAT, it is clear that there is a difference in target audience. But the uptake of ALAT by academic users could be an important project result. Even though UX has not been leading in the design of ALAT, the involvement of an HTI expert, as mentioned in *section 3*, will most likely have had a positive impact on the user experience. The following observations can be made with regards to the UX evaluation factors considered in [GaCo14]:

**Style**: AMAS’s style has been evaluated by asking the survey participants whether they liked the style or not. As the paper does not really give a clear definition as to what ‘style’ implies or how it is measured, it is difficult to form any relevant opinions regarding style in ALAT.

**Color**: The color scheme of ALAT has been kept calm and clean. Only a few colors are used to draw the interface (blue, white, black) in order to keep the interface uncluttered and calm. When comparing this to AMAS we can conclude that, though a different set of colors has been used, both interfaces use few colors. Bright colors and large contrasts have been avoided in both designs. Unfortunately the survey does not motivate its color scheme beyond the basic opinion of the participants.

**Look & Feel:** The survey on AMAS resulted in the following keywords to describe AMAS: ‘orderly, ‘calm’, ‘minimalist’, ‘functional’ and ‘focused’. Even though these keywords apply to ALAT as well at first glance, no surveys have been conducted to confirm this statement.

**Familiarity**: AMAS is evaluated in terms of familiarity by asking survey participants whether the interface reminded them of any other similar tools. The results are again hard to quantify and do not lead to more than a few statements and opinions. ALAT has been created using the Bootstrap[[2]](#footnote-2) framework in order to create familiarity. Bootstrap is a popular HTML, CSS and JS framework. It is used to create a uniform and familiar style throughout any web application.

1. My Online Teacher [↑](#footnote-ref-1)
2. http://getbootstrap.com [↑](#footnote-ref-2)