Information Architecture

The information architecture of an electronic store is important for user navigation, database management and scalability. A good information architecture should be modular, accommodate new groups and efficiently display and organise large amounts of data. In our toy car model, we use a hybrid organisational structure with macro content labels(groups), micro content labels(subgroups) and sorting to specify products further.

The content of our model are the toy cars themselves, with a volume necessitated by demand and supply. The data type of the cars will most probably be tables in an SQL database, possible to extract into JSON from the backend. The content of our IA model is divided into the following parts: business goals, funding, technology and resources. The final part of the model are the users themselves, who have the following attributes: user audience, specific tasks, needs and experience. For example, for toy cars we may have experienced professional model car builders or casual parents buying toys for their young children.

A well-planned model is a beneficiary solution for all of the people involved with the project. It allows users to find information quickly, and use a model + labels that suit varying needs, preferences and expertise. It also saves costs for the owner of the system – CPU cycles, maintenance costs, hard drive queries, power & heating bills. Ultimately a lacking user experience will lead to the loss of customers, bad reviews and declining market share.

# Inexact organisation schemes

Groups:

* Lego cars *– children & collectors*
* Small (hot wheel) cars *- children*
* Remote Control cars *– older children and teens*
* Model/replica cars *– hobbyists and collectors, teens*
* Self-built/painted cars *– hobbyists and collectors, teens*
* Small kid’s toy cars *– younger children*
* Collector/vintage cars *– hobbyists and collectors*

Sub-groups*:*

* Colour
* Size
* Brands
* Themes – *e.g. lego set themes*
* RC- *motor type (Electric, petrol, gas), speed, 4WD*
* Spare parts
* Accessories

# Exact organisation schemes

sorting

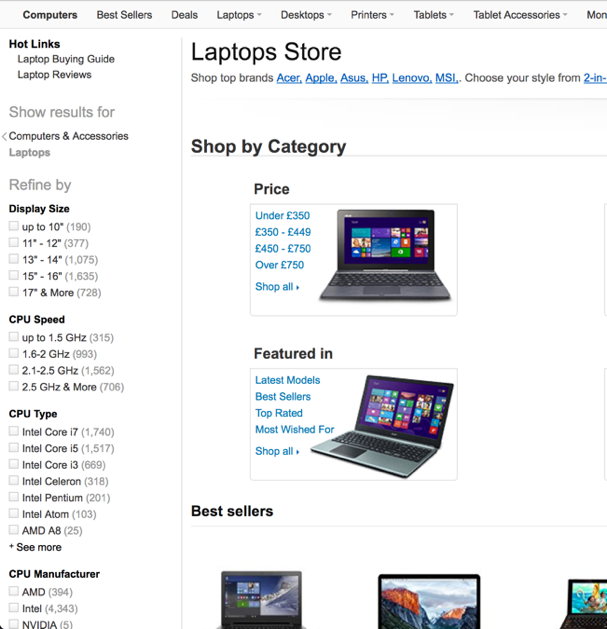
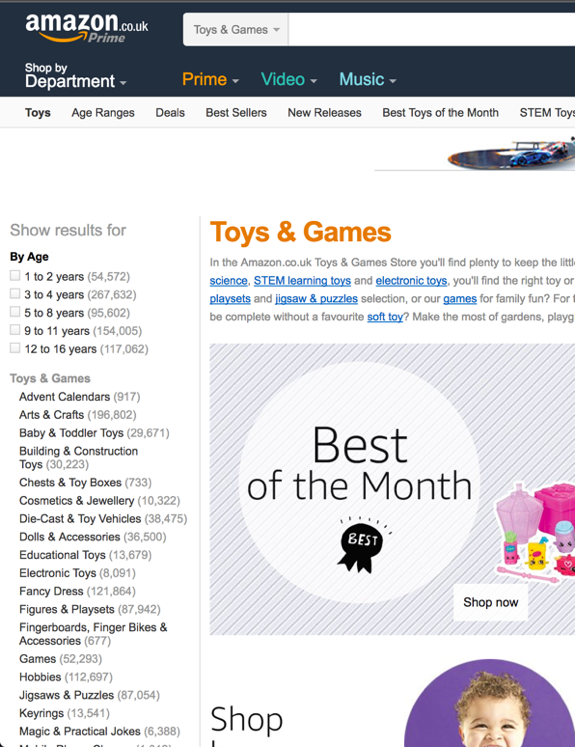
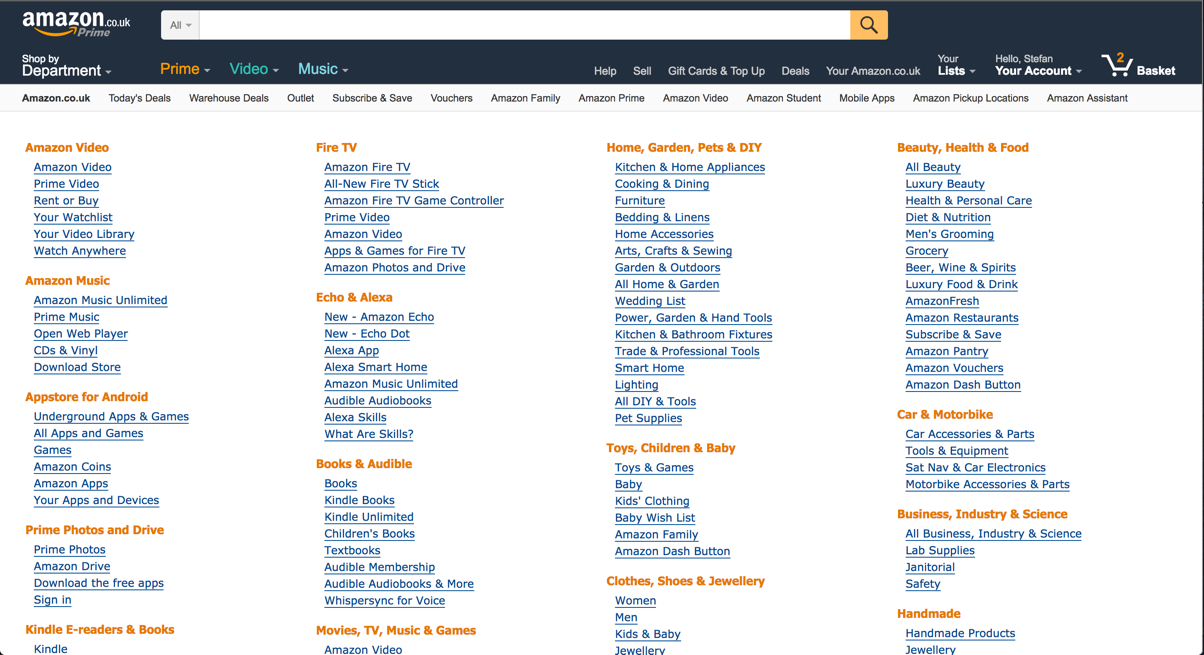
* Price *- ascending, descending*
* Alphabetical *– Ascending, descending*
* Release date (chronological) *– possibility to also show future releases*
* Customer review *– ascending according to rating*
* Age filter *–e.g.* *filter toys made for young children*
* Most popular – *most viewed cars e.g. in the past month*

With this IA model we hope to serve a variety of customers in an efficient manner. Next we will research IA models from a large existing company to see existing implementations.

# Research - Amazon

For research we decided to tackle the elephant in the room, amazon. With a massive track record they’ve put a lot of though into their information architecture, so we figured why not learn from the greats.

First Amazon sorts their products into departments, which are equivalent to our groups, the macro content labels. These allow customers to quickly filter their needs without being overwhelmed by a massive amount of sub-groups. The ordering of departments is also very important – they show a mixture of amazons own branding, accompanied by what Amazon believes most users are looking for.



Next, Amazon uses a mixture of sub-groups and sorting tools to refine the users search. Amazon have different sub-groups for each department, likely implemented through analysis of user activity. Although somewhat unnecessary for a small toy car company, analysis of user activity could be feasible with enough capital, developer skills and perceived results. For example, the Toys & Games department first has an age filter, then sub-groups, characters, delivery options brands and customer reviews. This makes perfect sense, since age is most likely the deciding factor on buying kids toys. In contrast, Amazon’s laptop department first sub-groups display size and price, proving the nature of Amazon’s adapted interface. With new open source A.I. and machine learning libraries becoming available, implementing adaptable labels and micro content labels is not completely out of the reach for medium-large size companies.

Amazon’s labelling systems are specific, clear, predictable and consistent. From the top of the two left most images above you see large labels for department groups with short names and clear messages. They are in present tense without articles to make them as clear as possible. In our menu (which is covered in another section of this report) we strive to do the same. Our labelling system also takes into account popular searches and adds sections accordingly for trending items – like Amazon’s Harry Potter Section. We also strive to keep labels consistent through our pages- although the ordering can change, the naming scheme will be consistent.

Hierarchy-wise, amazon’s user experience goes as follows (disregarding search):

Front page -> Possibility of trending items/user analysis -> Department -> category/micro content label/exact organisational scheme (as many times as needed)

-> product list -> selection of single item.

Our IA model goes from:

Front page -> Possibility of trending items/user analysis -> Groups -> Subgroups -> More subgroups/exact schemes -> Product list tailored for user -> selection of item.

In conclusion, we hope to replicate the behaviour of Amazon for our toy car model using groups, subgroups and sorting schemes.

Analytical Usability Evaluation

Analytical usability evaluation is important to test and stress our model systematically, collecting results as we go on. We want a system that is modular, simple to use and effective – even when unexpected things happen. We will use three methods for usability evaluation: First we will assess our accessibility with cognitive walkthrough, then use a keystroke-level model to test the efficiency of our site and finally a Heuristic evaluation to assess expert users.

Our prototypes and page design take into account all scenarios for special cases, such as users with disabilities. Our plan is to use aria in HTML to provide text equivalents for images and audio- making sure the aria tags are as descriptive as possible. Our menus, navigation and labels do not use colour to convey meaning, and tables are clearly laid out line-by-line for screen readers. Especially things like ratings – with stars – are often overlooked, so we made sure everything is included in aria. When it comes to features like payment and shipping, text-to-speech and password managers are available for maximum accessibility.

From an accessibility standpoint, the cognitive walkthrough would go as follows:

1. Enter the site
2. Use a screen reader to understand page contents – optimize for search bar & departments
3. Select the search bar/department
4. Use keyboard/and or vocal recognition to enter a query
5. Screen reader reads the result of the page – optimizing content and not header, nav etc.
6. Select product
7. Add to cart – make cart, checkout etc. well labelled for aria
8. Go to cart & checkout
9. Choose payment method & postage
10. Provide billing information
11. Finalize purchase

## Critical evaluation

Search has to be well optimized to handle the user request efficiently and show related products. It also must handle typos, suggest proper spelling and show contents regardless. Especially for visually impaired people the results must be clear for screen readers and not cluttered with information.

Product list must be clear and provide correct amount of information. For example, Amazon’s product lists show shipping information, user ratings etc. This information can be found on the product page, as screen readers are slow showing this information in the product list makes site navigation extremely cumbersome for customers with visual impairments. Also the list must exclude products that are not available in the country or are out of stock – these are usually indicated by visual feedback which are difficult to convey with screen readers.

Ads must be hidden from screen readers. We also consider optimization of header, navigation bar once the user is already on a product page. For example, when a user first lands on a site the navigation bar is useful information. However, once the user is navigating through products, it becomes less important and making it read every site would simply become a nuisance.

We outsource some tasks external companies who use tools such as Bobby to ensure maximum accessibility.

# KLM Model

For our KLM evaluation we test how long it takes a user to find their desired product- a toy car.

1. Move hand to keyboard (H – 0.4s)
2. Enter URL and press enter (K – 3s)
3. Page load (R – 0.2s)
4. Move hand to mouse (H – 0.4s)
5. Find search bar (M – 0.2s)
6. Move mouse to search bar (H - 0.4s)
7. Click search bar (B- 0.1s)
8. Move hand to keyboard (H – 0.4s)
9. Enter desired search (K – 2s)
10. Press enter (K – 0.1s)
11. Page load (R – 0.2s)
12. View contents of the page (M – 1s)
13. Fine-tune search (optional) (K – 2s)
14. Compare products (optional) (M – 3s)
15. Move hand to mouse (H – 0.4s)
16. Scroll and find desired product (B – 3s)
17. Click product (P, B – 2s)

Total time – 13.8 seconds

With options – 18.8 seconds

While this time may vary greatly per user, it gives us a good indication on the effectiveness on our site’s layout and clarity. By having a ‘baseline’ user experience we can test out different interfaces and compare them using real user data, finding the optimal solution in minimal time. Obviously a screen reader would take much longer for these tasks, and in contrast a tech-savvy user who already knows what he/she is looking for will be much faster. We strive to make our site appeal for all kinds of users, also the older generation who prefer large fonts and visual elements for easy navigation.

# Heuristic Evaluation

For our heuristic evaluation, we used usability experts – first letting them perform the evaluation on their own. Then letting them go through the interface again to focus on specific interface elements. We used the popular Nielson’s heuristics for evaluation.

1. **Visibility of the system status -**  To maximize the page presence and keeping the user informed on what’s going on, we have a fixed navigation bar on the top of the page, titles that explicitly state on which page they are on and helpful cues such as ‘go to the top’ at the bottom of a product list. We use different colours for banners at different sections of the page, and links to other pages are marked clearly.
2. **Match system and the real world** – We used real world conventions for e.g. going back in a stack of queries, using intuitive buttons for forms and using flash messages and modals effectively. Our words and phrases for navigation are in clear present day English.
3. **User control and freedom -**  To make the user be in control we made sure to always add a clear exit to the main page and especially from the shopping cart. While it is a great marketing strategy, when a user is making a purchase we want the user to always see a clear path of exit in case they change their mind. Confirmation and availability of cancelling once an order is complete is essential for achieving this.
4. **Consistency and standards** – To achieve consistency we used the obvious methods, shared CSS files for most pages, same fonts and navigation styles. A button will always mean the same thing no matter which page the user is on.
5. **Error prevention –** to minimize errors we used front-end and backend validation, immediate user feedback and rules on our database implementation – transactions have unique ID’s and unique customer ID’s which cannot be duplicated etc.
6. **Recognition rather than recall –** Users are followed through their cookies, which remembers their viewed and favourite toy cars even without signing in.
7. **Flexibility and efficiency of use –** We have JQuery setup to provide immediate suggestions for departments and products in the search bar, as well as optimized search paths described in the KLM model.
8. **Aesthetic and minimalist design –** while we strive for minimalism we don’t take it overboard like the hipster front end devs like to do nowadays. Function over form with a pinch of minimalism.
9. **Recover from errors -**  as described in the error prevention, we have exceptions setup for all test cases with informative user feedback and state-of-the-art prevention of data loss.
10. **Help and documentation**
11. **Privacy –** To ensure user privacy we are based in an undisclosed offshore country that is out of reach of CIA, NSA and any federal government agencies. Users have options to pay with bitcoin and we enforce the use of HTTPS and secure SSL certificates to make transactions untraceable if the user wishes to do so. We also offer fake boxes for our toy cars to hide them from border & postal inspections. For optimal user privacy we recommend using Tails OS and Tor which comes bundled with it along with a VPN of your choice. In terms of security we have a Hackathon and Bugbounty programs setup, along with outsourcing pentesting to the best in the industry.

## Critical Evaluation

Add some improvements here-

• Appropriateness of method(s) chosen

• Good quantity of work if bigger group (e.g. four people in group, than try multiple methods)

• If doing Heuristic Evaluation, in bigger group use multiple evaluators

• Good defence of why something is good (if you cannot find problems)

• Could also evaluate somebody else’s system

• Tell me exactly where the problems are, and maybe even how to improve them