

Investing in product success through careful browser selection

Ekioh Ltd, Cambridge UK.
flow@ekioh.com

Introduction

It is hard to believe that it's over 20 years since the first commercial IPTV networks were launched. Developed in-house by pioneering operators, these deployments quickly gave way to third party middleware based solutions as the technology made its first tentative steps towards maturity. Middleware companies recognised that, in order to address the wider market in a cost effective way, they needed to separate the user interface from the service logic and set top box hardware so many looked to markup languages from the wider internet as the solution.

Whilst other markup languages have been very successful, the wide range of development tools and large pool of low cost developer talent have firmly established HTML as the markup language of choice for the TV market. The availability of great development tools has also led to a broadening of the application authoring community to include those with creative design skills. This has resulted in a marked improvement in the visual quality of user interfaces and the look and feel of applications. In many cases the user interface has become a point for service differentiation and customer retention.

IPTV itself has also changed dramatically. Initially the market followed the traditional broadcast TV model and supplemented that with on-demand services like VoD, but the success of catch-up TV services and OTT operators have created a far more diverse market. Also gone are the days where the operators had total control over the hardware their customers used, replaced by a model where more often than not, the customer provides the hardware themselves. The proliferation of hardware has led to a thriving embedded browser market which is served by many specialist providers who mostly offer tailored versions of mainstream internet browsers like WebKit and Blink.

Contrasting Use Cases

Mainstream browser development is, in the most part, driven by the 'surfing' needs of the mobile phone and desktop computer market. Bodies like the WHATWG community adopt a "living standard" philosophy where the market is used to determine the benefit of new features. Those which are not embraced by content developers are replaced or withdrawn from the standard. This living standard is reflected in the operational lifecycle of surfing products which incorporates relatively frequent browser software updates.

By contrast, for consumer products like set top boxes and smart TVs, the browser is embedded as a foundation component upon which higher level functionality is built rather than being an application in its own right. In-life changes to foundation components can cause the end user the inconvenience of being unable to use the product during the software update and carry the risk that any device for which the update fails could be rendered useless. As a result, updates are infrequent and so choosing the right components at the outset is vital.

Benchmarking

The factors affecting browser selection in the embedded market differ greatly from those of the surfing market. For desktop browsers, a rich feature set and the ability to have multiple pages open at once are normally the top priorities whereas in the embedded market, single application performance on resource constrained platforms and longevity are paramount. These differences in selection criteria are further complicated by the lack of suitable benchmarking tools.

It's not that benchmarking tools are in short supply. A simple internet search will reveal dozens of browser benchmarking tools, but without a complete understanding of what needs to be compared, these tools are more likely to drive an incorrect choice than they are to help narrow the field of suitable products.

Some benchmarking tools focus purely upon JavaScript performance and in the most part measure JIT performance. JITs compile JavaScript into machine code at run time and are commonly tuned to produce best results on high performance platforms. The compilation overhead is insignificant when the same script is run many times, but can dominate on slower platforms, particularly when a script is run infrequently. On lower performance hardware such as is common in consumer electronic products, popular applications like YouTube TV actually run more slowly if the JIT is enabled.

In the surfing market JavaScript benchmarking makes reasonable sense as the vast majority of websites designed for mobile and desktop use are built using third party JavaScript libraries which dramatically reduce development time, but greatly increase the amount of JavaScript being used. Analysis of the processing load from a sample of the Alexa top 500 websites shows that most spend more than half their time processing JavaScript.

Whilst 3rd party libraries are often used in the embedded market, well crafted user interface applications regularly only spend around a third of their time on JavaScript operations, being much more heavily weighted towards layout and painting tasks. Embedded applications also rarely employ the levels of repetition necessary to overcome the JIT compilation overhead making most JavaScript benchmarking figures relatively meaningless.

The headline numbers produced by feature driven benchmarks are similarly biased towards the surfing market. Scoring is generally weighted towards new features and often includes 'pre-standards' and processor intensive features. For the embedded market, longevity is key and so any feature which isn't mature is unlikely to be welcome. The constrained nature of most embedded platforms also rules out the use of more resource hungry features.

Choosing your Selection Criteria

The specialist nature of the embedded market has resulted in a reasonable number of browser suppliers. Many share common WebKit/Blink roots with those on the desktop such as Safari and Chrome, but most have been optimised for embedded use. Given the lack of suitable benchmarking tools, making the right choice for a specific application is no easy task, but a structured and methodical selection process can lead to significant benefits downstream.

Firstly, choosing a browser because you're familiar with its desktop counterpart isn't a good shortcut through the logical selection process. The huge differences between a desktop and the target platform/use case mean that simply selecting an embedded browser based upon familiarity from unrelated products is unlikely to be the optimal choice.

Feature scoring benchmarks can be a useful tool in browser selection providing the results are scrutinised. Almost all feature benchmarks back up their headline scores with individual feature analysis making it possible to create a 'useful score' excluding those features which are immature or require greater hardware performance than is economically viable for the target market. It should also be noted that in some cases, benchmarks merely check for the presence of a feature and not the degree to which it's implemented making it possible for some browser providers to manipulate their scores. In the most part such manipulation is reserved only for features which are rarely used but this does reinforce the point that headline scores can be misleading.

Application development communities will always want the richest feature set possible and the tendency to convey this through requests for high feature benchmark scores is great. Working closely with your chosen development community so they understand the capabilities of the target platform can help create the 'useful score' feature list which can be used as the basis for browser comparison. Such communities will also value the availability of off-target development environments and content debugging tools as these provide a very clear indication of the final product's capabilities.

About Ekioh

Ekioh designs and develops rendering software for a wide range of consumer products and resource constrained platforms. The company's products provide the high performance, compact footprint and flexibility necessary to address the wide range of application and UI rendering challenges presented by the consumer electronics and embedded systems industries. Best known for its TV and set top box solutions, Ekioh browsers are deployed in tens of millions of products in over 30 countries around the world.

Ekioh's engineering team brings together expertise in graphical systems, embedded software, set top box, TV silicon and robust software design. Using this combination of skills, fuelled by intense customer focus, Ekioh's product quality and customer support are second to none.

Having a very clear idea about the applications that the browser is to support and using that to create benchmarking criteria may sound like an onerous task, but some upfront investment can generate significant cost and time savings later on. Benchmarking on hardware which is representative of your target hardware is also highly advisable as a browser that has been optimised for one hardware configuration may perform very differently on another.

Tailored use case benchmarking can also provide benefits beyond the selection of the most suitable browser. Understanding how performance might affect your product's usability can enable the development of key performance indicators (KPIs) for critical user interactions. Benchmarking of these KPIs, such as the time to move between EPG screens and switching times between different OTT apps, can help ensure the product stays on track throughout its development.

It is also worth remembering that users' performance expectations tend to increase over time so knowing the margin by which your platform exceeds today's KPI figures can be used to determine its probable longevity. If the KPIs suggest that the product's lifespan is likely to be less than required, changes to the UI design, the hardware or the in-life upgrade strategy can all be considered before product launch whilst the cost of change is relatively low.

Summary

Making the right browser selection can have a significant impact on a product's success, development time and longevity. The upfront investment to really understand the needs of your market and to use those needs to create a set of tailored benchmarks covering browser feature requirements and on-target performance is money well spent. Using these criteria to direct the browser selection process backed up by carefully selected KPIs to ensure user experience remains in focus during the development process are essential steps in the creation of products which customers will value and love for years to come.