SuperSlob: the development of a parametric component jointing regime for standard sheet materials.

Ryan Tubby¹, Richard Burnham¹ and Robin Green¹

¹University of Tasmania, Launceston, Australia

ABSTRACT: The Castle, a collaboration between the School of Architecture & Design (UTAS) and Youth Futures Inc (YF), assists youth at risk of homelessness by deploying micro-dwellings to households experiencing spatial and emotional distress. The digital construction system has been tailored specifically for low-skilled labour through reliable and predictable component jointing, a limited palette of materials, minimal tooling and assembly instructions. A detailed interrogation of the construction ‘grammar’ has resulted in the development of a software plugin that increases the efficiency and reliability of the design, documentation and fabrication process. The plugin, called ‘SuperSlob’ (the term ‘slob’ derived from the slot and tab connection) has been developed by students and staff at UTAS, in partnership with a software programming consultant.

The brief for SuperSlob has prioritised ‘accessibility’ so the Castle construction system can progress from limited ‘customised production’ toward mass customisation. SuperSlob integrates all stages of design and documentation – from module and material parameters, to component configuration, ‘router-ready’ jointing, and file management. SuperSlob is currently being tested by students in two ‘Learning by Making’ studios: a ‘Crisis Accommodation Bus’, and ‘Design Island’. Feedback from the users of the plugin will contribute to the further development of SuperSlob.

Conference theme: Design education and computing
Keywords: digital design, CAM, plywood

INTRODUCTION

Initially, The Castle was aimed at assisting youth at risk of homelessness by deploying micro-dwellings to households experiencing spatial and emotional distress. The brief specified a dwelling that was small, mobile, autonomous and spatially clever. Four prototypes (C1, C2, C3 and C4) were designed, fabricated and constructed by B.Env.Des. (first to third year) architecture students in collaboration with Youth Futures Inc. (YF). The fifth constructed prototype (known as C5) has proceeded to serial manufacture by long-term unemployed trainees in a Federal government funded training program run by YF. Social aspirations remain the main focus of this collaboration (Burnham and Green 2011a), but the introduction of a digital design tool has broadened the scope of possibilities, such as, an exhibition display environment, a bus fit-out, and equipping a pair of shipping containers as a ‘Mobile Castle Factory’.

Development of the construction system has benefitted from these applications, and consequently this directly benefits the development of the responses to the initial brief.

A detailed interrogation of the construction ‘grammar’ has resulted in the development of a software plugin that dramatically increases the efficiency and reliability of the design, documentation and fabrication processes. The plugin, called ‘SuperSlob’ (the term ‘slob’ derived from the slot and tab connection) has been developed by students and staff at the School of Architecture & Design (UTAS), in partnership with a software programming consultant. SuperSlob assists low-skilled labour to assemble Castles through the reinforcement of reliable component jointing regimes, and retention of minimal tooling requirements.

Observations of a gap emerging in digital design between assemblers and designers has informed the workflow of the plugin, and choice of the SketchUp platform. It is predicted this gap will only continue to become wider as design systems become increasingly sophisticated, specialised and time intensive. Accordingly, the brief for the plugin has prioritised ‘accessibility’ in the expectation that the Castle can progress from a limited ‘customised production’ toward mass customisation where (as envisaged by Alvin Toffler) a future consumer could become involved in the design and manufacture of products made to their individual specification (1980), to close the assembler/designer gap.

This paper documents the development of digital design tools throughout the evolution of the Castle. The paper focuses on the inception of the plugin, and its implementation in a fifth year ‘Advanced Design Research’ project - Design Island (DI). Beyond the DI project, SuperSlob has now also been applied to a bus fit-out (B1) and is currently being used for the design of a Mobile Castle Factory (F1). Strong interest from both educational and commercial design sectors indicates the construction system and software has potential beyond the initial Castle brief.

The paper proposes that after several previous attempts to effectively apply digital design and fabrication to intensive design and build projects, SuperSlob satisfies three desired characteristics: being easy to learn, reliable and predictable, and sufficiently flexible for application to a variety of design scenarios. The continual and iterative nature of software development suggests that the results of these early trials will lead to several rounds of improvement before the eventual release of the plugin. Interest in SuperSlob from both educational and commercial ends of the design spectrum however, raises questions about the type of release.
1. EVOLUTION OF CASTLE CONSTRUCTION GRAMMAR

The Castle has always had an underlying agenda of questioning what a home needs to be. At around 10m² the Castle is a small fraction of an average Australian dwelling. However, they do satisfy criteria required for a sense of home: security, public and private identity, defined threshold, space for exhibition of personal belongings and spatial separation of bathing, cooking, relaxing and sleeping (Burnham and Green 2011b). The Castle also questions practices of mainstream construction. The brief for the construction system had to satisfy four basic principles: an efficient use of a renewable resource, low weight, applicability to a low-skilled workforce and that it offered opportunities for mass customisation (Burnham and Green 2011a).

To understand SuperSlob's origin it is important to relate how the digital design content of the Castle has emerged alongside the evolution of the construction system (see Figure 1, and Table 1). With many Castle prototypes designed and constructed by groups of twenty or so students in fourteen day ‘Learning by Making’ (LBM) studios, how effectively the construction system can be understood and used by students is an important consideration.

Castle 1 (B.Env.Des. LBM: C1, test rig and component research) created an adjustable rig for spatial studies relating to micro dwellings (see Figure 2). The rig was assembled from bolted pine framing and honeycomb infill panels assembled from digitally cut cardboard.

Castle 2 (B.Env.Des. LBM: C2, design, fabrication and construction), the first habitable iteration, employed a modular, prefabricated stressed skin plywood panel system (see Figure 2), which on later analysis was deemed to have failed in terms of weight, accuracy and replicability. An internal grillage storage wall was digitally designed and fabricated using a Vectorworks VectorScrip 'Box Tool' written by UTAS colleague, Justin Beall. This was the initial tentative step into parametric design for the Castle (Burnham and Green 2009).

Castle 3 (B.Env.Des. LBM: C3, design, fabrication and construction) was the first attempt to apply digital design and fabrication to an entire building. The system, named ‘panitectute’ (a hybrid of ‘panel’, ‘furniture’ and ‘architecture’) was inspired by teardrop caravans, kitchen cabinetry and ‘Furniture House 1’ by Shigeru Ban (1995). The panitectute panel was a folded-plate, resembling a drawer assembled from five digitally cut components (see Figure 2). This utilised an improved version of Beall’s Box Tool and a substantial amount of ‘manual’ drafting in Vectorworks. Wherever possible, panel flanges (the ‘drawer’ sides) were extended internally and integrated into furniture such as stairs, benches or storage cabinets. Panitectute facilitated minimal tooling comprising: a rubber mallet, glue gun and a battery powered driver, thereby mimicking the methodology of kitchen cabinetry and flat-pack furniture. Conventional cladding and flashing proved, within the aspirations of the brief, expensive and time consuming. (Burnham and Green 2011a).

Castle 4 (B.Env.Des. LBM: C4, design, fabrication and construction) applied panitectute to SketchUp Pro 7 turning the basic drawer composition into a ‘dynamic component’ (a native function of the ‘Pro’ version of the software). The panel’s characteristics could be manipulated through a dialogue box, allowing alterations to the overall dimensions, the option of a window, and adding an extra stiffener beyond a specified dimension (see Figure 2).
Castle 5 (Training program at YF: production of twelve C5s, currently ongoing, full completion early 2013) was a distillation of lessons learnt from previous prototypes. The main development was to move from a panel system to a fully integrated construction, where every component was doing a job of either enclosing space or contributing to built-in furniture – eliminating components being used for structure only (Burnham and Green 2011a). Components were designed and slobbed in Vectorworks where ‘strings’ of slots and tabs were manually added to the component. Despite numerous scale models alignment issues continued through to full-scale production, and were a source of frustration for YF trainees (Burnham and Green 2011a).

Robin Green and Richard Burnham had investigated a number of software alternatives by this stage, with the specific aim of improving the reliability and predictability of the slobbing process. Cabmaster, a kitchen industry package that automatically applies jointing patterns to panels of specified parameters, has excellent production credentials but was constrained by a complex interface and the high cost. The C6 LBM students were trained in Solidworks, deemed to be a good balance between production and design as well as having parametric potential. Unfortunately learning Solidworks consumed most of the studio time, to the extent that not even a complete scale model was produced. At this time we questioned whether “continued perseverance with digital technologies for the Castle, however sophisticated, will inevitably be at the expense of the experience of the LBM studio” (Burnham and Wallis 2012:196).

The ‘slobbing table’, a 150/12mm tartan grid based in Vectorworks improved the design and documentation process. Components whose dimensions complied to the restrictions of the grid were operated on by applying the appropriate jointing pattern from elements around the table’s perimeter (see Figure 3). Models for C7 and C8 were built using the slobbing table, resulting in substantial improvements in joint alignment. The design process remained somewhat cumbersome, with a raw 3D model being used for initial component configuration and 2D for joint application.

The bri...
accessibility and reduce this gap by simplifying the design process, and by decreasing the specialised and time consuming nature of the documentation procedures.

### Table 1: Digital application to the Castle

<table>
<thead>
<tr>
<th>Castle</th>
<th>Extent of Digital Application (“Physical:Digital” Work Ratio)</th>
<th>Program</th>
<th>System</th>
<th>Slobbing</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>(none)</td>
<td>(none)</td>
<td>(none)</td>
<td>(none)</td>
</tr>
<tr>
<td>C2</td>
<td>25% minor</td>
<td>Vectorworks</td>
<td>S.S.P.P.</td>
<td>manual</td>
</tr>
<tr>
<td>C3</td>
<td>50% even</td>
<td>Vectorworks</td>
<td>Paniture</td>
<td>manual</td>
</tr>
<tr>
<td>C4</td>
<td>60% substantial</td>
<td>SkUp, VW, AutoCAD</td>
<td>Paniture</td>
<td>manual</td>
</tr>
<tr>
<td>C5</td>
<td>60% substantial</td>
<td>Vectorworks</td>
<td>tartan modular</td>
<td>semi-auto</td>
</tr>
<tr>
<td>C6</td>
<td>95% excessive</td>
<td>SolidWorks</td>
<td>tartan modular</td>
<td>semi-auto</td>
</tr>
<tr>
<td>C7</td>
<td>80% excessive</td>
<td>Vectorworks</td>
<td>tartan modular</td>
<td>semi-auto</td>
</tr>
<tr>
<td>C8</td>
<td>85% excessive</td>
<td>Vectorworks</td>
<td>tartan modular</td>
<td>semi-auto</td>
</tr>
<tr>
<td>D1</td>
<td>40% optimised?</td>
<td>SuperSlob</td>
<td>SuperSlob</td>
<td>SuperSlob</td>
</tr>
<tr>
<td>B1</td>
<td>40% optimised?</td>
<td>SuperSlob</td>
<td>SuperSlob</td>
<td>SuperSlob</td>
</tr>
<tr>
<td>F1</td>
<td>(grid indicates percentage)</td>
<td>SuperSlob</td>
<td>SuperSlob</td>
<td>SuperSlob</td>
</tr>
</tbody>
</table>

Source: (Author 2012)

The plugin, developed over an intensive six week period by staff, students and the SketchUp programming consultant, is basically 3D in-situ panel modeling, followed by the slobbing of component connections, and export of fabrication ready files. The workflow contains as much of the design and documentation process as possible; overall design of the object, edge detailing, and file export are all managed by SketchUp. SuperSlob is driven by the construction grammar that has resulted from the interrogation of the modular system, developed by the slobbing table. A dialogue box of settings controls the predictability and reliability of the physical construction joints, their tolerances and the consistency amongst multiple drafting authors within the design team. Either a drop-down menu, or a floating palette of the available tools conducts the operation of SuperSlob. These tools are adaptable and expandable, allowing them to include new jointing patterns. Slobbing time is reduced due to removal of the need to manually draft component connections, and the level of customisation has been increased by the versatile nature of the newly defined construction grammar, possible of building anything that joins two planes.

### 3. HOW SUPERSLOB WORKS

#### 3.1 Mechanics of the Plugin – Overview

Whilst SketchUp has been chosen for its simplicity, to achieve our goals for this digital tool we have had to compromise the accessibility of the tool. The free version of SketchUp does not provide the same functionality of the Pro version, which unfortunately requires the purchase of a license. SketchUp Pro 8 is currently the only version of the SketchUp suite that can perform solid Boolean operations, essential to the current mechanics of the plugin.

Creating flat coplanar ‘proxies’ in 3D space facilitates the in-situ panel modeling. These proxies form the 3D ‘centreline’ of a panel defined by the specified material thickness. Once a series of panels have been arranged to form an object the panels can then be ‘slobbed’ by detailing the panel edges. A pop-up menu provides a suite of options to either ‘slot’ or ‘tab’ the selected edge (see Table 2). Once an option is chosen, SuperSlob arrays a series of scaled rectangular prisms along the length of the edge to subtract the slots, and union the tabs. SuperSlob then arrays a scaled cylinder to subtract the internal corners for ‘mouse ears’ (Burnham and Green 2011a) (see Figure 4).

![Figure 4: Function of the ‘Mouse-Ear’](source)

All of these operations are performed by equations based on four specifications in the settings dialogue box: material thickness, module size, router bit size, and tolerance. Although these equations mean nothing without a reference point, so the origin point of each panel provides it, specified by the first clicks when creating each panel.

#### 3.2 The Tools

Initially SuperSlob consisted of six tools, but as we amended and expanded the capabilities of the plugin we dropped some tools and added new functions, resulting in the current selection of eight tools. In the following sections the tools and their operation are explained in more detail.
3.2.1 ‘Settings’
Allows specification of customisable settings in a dialog box. The options are as follows:

- **Module**: default 150 mm or other module
- **Thickness**: default 12 mm or other thickness
- **Pilothole diam**: default 3 mm or other diameter
- **Router diam**: default 4 mm or other diameter
- **Tolerance**: default 0.2 mm or other dimension
- **Text height**: default 36 mm or other height
- **Text depth**: default 1 mm or other depth
- **Text starts**: default “001” or other number
- **Sheet length**: default 2400mm or other size
- **Sheet width**: default 1200mm or other size

3.2.2 ‘Make Panel’
Activation of the Make Panel tool hides the ‘panel’ view and reveals only the proxies in ghosted x-ray style. This allows controlled creation of panels ‘on-grid’. The first picked point sets the panel’s origin, with the possibility to ‘snap’ to other proxies’ corners as desired. After the first point is picked a modular horizontal grid appears, centred on that point and the dimensions of the panel are constrained to modular lengths along the x/y axes. On picking the second point, a second grid appears perpendicular to the first. Selection of the third point, determines the plane of the panel and the unnecessary grid is removed. The selection of more points continues in the panel's plane until clicking on the start-point again, which completes the panel. The panel component is named “Panel#001”, “Panel#002” and so on, in incrementing fashion. Within the panel there are three ‘groups’; the proxy, the panel itself extruded to the set thickness centred on the proxy, and a centralised component of text showing the panel's reference (e.g. ‘001.’).

3.2.3 ‘Edge Detail’
This, the primary slopping tool, allows the detailing of a preselected panel’s edges. An edge is highlighted in orange with ‘X’ markers at its ends. The <Enter> key selects the highlighted edge while the <Tab> key moves to the next edge. On <Enter> a dialog box lists the types of edge-details that can be applied (see Table 2).

<table>
<thead>
<tr>
<th>Name</th>
<th>Profile</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>slot</td>
<td>![Slot Profile]</td>
<td>Standard set-out (half module size): thickness x (module ÷ 2) Slot with mouse-eared inners, pilot holes, and tolerances.</td>
</tr>
<tr>
<td>tab</td>
<td>![Tab Profile]</td>
<td>Standard set-out (half module size): thickness x (module ÷ 2) Tab with mouse-eared inners, pilot holes, and tolerances.</td>
</tr>
<tr>
<td>narrow_slot</td>
<td>![Narrow Slot Profile]</td>
<td>Standard set-out (quarter module size): thickness x (module ÷ 4) Slot with mouse-eared inners, pilot holes, and tolerances.</td>
</tr>
<tr>
<td>narrow_tab</td>
<td>![Narrow Tab Profile]</td>
<td>Standard set-out (quarter module size): thickness x (module ÷ 4) Tab with mouse-eared inners, pilot holes, and tolerances.</td>
</tr>
<tr>
<td>tab_internal</td>
<td>![Tab Internal Profile]</td>
<td>Similar to 'tab', but has no pilot holes. For use in a full 'slot_internal'.</td>
</tr>
<tr>
<td>tab_internal[pocket]</td>
<td>![Tab Internal Pocket Profile]</td>
<td>Similar to 'tab_internal', but for use in pockets 'slot_internal'[half-thickness].</td>
</tr>
<tr>
<td>dovetail_slot</td>
<td>![Dovetail Slot Profile]</td>
<td>Female dovetail (half module size): thickness x (module ÷ 2) Has slots for short-tabs, mouse eared-inners and pilot holes.</td>
</tr>
<tr>
<td>dovetail_tab</td>
<td>![Dovetail Tab Profile]</td>
<td>Male dovetail (half module size): thickness x (module ÷ 2) Has slots for short-tabs, mouse eared-inners and pilot holes.</td>
</tr>
<tr>
<td>short_tab[dovetail]</td>
<td>![Short Tab Dovetail Profile]</td>
<td>Creates 'short tabs' with no pilot holes, to support a pair of dovetailed edges.</td>
</tr>
<tr>
<td>plus_half</td>
<td>![Plus Half Profile]</td>
<td>Removes all detailing back to an edge outset ½ material thickness.</td>
</tr>
<tr>
<td>minus_half</td>
<td>![Minus Half Profile]</td>
<td>Removes all detailing back to an edge inset ½ material thickness.</td>
</tr>
<tr>
<td>none</td>
<td>![None Profile]</td>
<td>Removes all detailing back to a flush modular edge (aligns with the centrelines/proxies).</td>
</tr>
</tbody>
</table>

Source: (Author 2012)

3.2.4 ‘Internal Edge Detail’
This is very similar to the ‘3.2.3 Edge Detail’ tool, although it only works on ‘sets’ of internal-edges around ‘cut holes’ such as windows. On activation of the tool, if there is more than one hole, the perimeters are highlighted in turn. Pressing <Enter> selects a hole, and then the edges of that hole are highlighted as in the ‘Edge Detail’ tool. This tool has the same options as the ‘Edge Detail’ tool (see Table 2).
3.2.5 ‘Add Internal Slot-Line’
This function allows a ‘slot-line’ to be drawn on to a selected panel, for use in making an array of internal details (see 3.2.6 ‘Line Detail’). On activation of the tool, the view automatically swaps to reveal only the proxies and a modular grid is placed over the selected panel’s proxy. To start an internal slot-line, the cursor snaps to a module interval or corner of another proxy’s edge. The selection of the end must also be on an orthogonal modular interval. The created line is then added to the contents of the panel for use with the ‘Line Detail’ tool.

3.2.6 ‘Line Detail’
This is also similar to 3.2.3 ‘Edge Detail’ tool, although this will only work on ‘internal slot-lines’ added to define perpendicular junctions. The available internal slot-lines are highlighted in turn. Table 3 outlines the list of possible internal slot-line details.

<table>
<thead>
<tr>
<th>Name</th>
<th>Profile</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>slot_internal</td>
<td></td>
<td>Standard set-out (half module size): thickness x (module ÷ 2) Internal slot with mouse-eared inners, pilot holes, and tolerances.</td>
</tr>
<tr>
<td>slot_internal[pocket]</td>
<td></td>
<td>Standard set-out (half module size): (thickness ÷ 2) x (module ÷ 2) Internal slot with mouse-eared inners, pilot holes, and tolerances.</td>
</tr>
<tr>
<td>halved_up</td>
<td></td>
<td>Halving joint with slot at the start of the internal slot-line: includes mouse-eared inners, and tolerances.</td>
</tr>
<tr>
<td>halved-down</td>
<td></td>
<td>Halving joint with slot at the end of the internal slot-line: includes mouse-eared inners, and tolerances.</td>
</tr>
</tbody>
</table>

Source: (Author 2012)

3.2.7 ‘Export Panels’
This function exports all the panels in SketchUp named “Panel#xxx” that have an instance placed in the model. Each panel is exported as a DXF in turn (in alphanumerical order), and placed into a folder in the same location as the SketchUp model, named: “ModelName_DXFsv”. The native SketchUp DXF export options are used, so they need to be [re]set as desired. A dialog box offers options to include the panel's “text tags” and whether to export the DXFs as 2D or 3D geometry. For 2D DXFs the geometry is flattened from the top down, then split into several layers per panel, each with a different purpose and colour in the DXF for simple processing at the CNC router.

3.2.8 ‘Re-Reference’
Re-reference offers the capability of renumbering a selected panel. For example: “Panel#001” > ‘100’ > “Panel#100” (typical formatting has been utilising three digits with leading zeros, although any length is possible). The panel is renamed, and the text group within it is also amended to suit the new reference.

4. APPLICATION OF SUPERSLOB

4.1 Design Island – Project Description
Design Island (DI) was the first project to incorporate the use of SuperSlob (see Figure 5). DI was a M.Arch. (fifth year architecture students) ‘Advanced Design Research’ project, in which a student group (the design team) designed, fabricated, constructed and deployed the final structure. The main objective of the project was to create a ‘showcase’ for 20 Tasmanian designers and their 26 objects for the 2012 Design:Made:Trade (DMT) exhibition in Melbourne.

Specifically, the students were required to produce, and deploy a 60 m² ‘display environment’ for a design exhibition. As this commercial agenda was notably different from the previous social applications of the Castle, the project was subsequently used as a chance to explore how SuperSlob worked as a process of designing and fabricating Castles (and other products able to utilise the Castle construction system).

A brief was determined in response to the supervising lecturer’s initial brief, discussions with the Design Island management body, and during several team meetings. As a result, DI had five primary aims: display, sell, unify, adapt, and represent island. These aims were manifested through the project by constructing a coherent, logical and memorable display environment that unified the attending Tasmanian designers, whilst also maintaining the ability to respectfully display the diverse range of objects in scale, quantity and identity. It also showcased the flexible (and customisable) nature of the Castle construction system, which affords DI several degrees of adaptability and flexibility allowing it to exist in varying scales and arrangements for future exhibitions. It also had an outstanding ability to sell these objects, as demonstrated by the $14,000+ taken over the four days of the exhibition. All this, whilst displaying and representing themes of ‘island’, an especially important task when DI was interstate.
4.2 Teaching and Learning SuperSlob

The final year M.Arch. student design team had no previous exposure to SuperSlob; many had never constructed a Castle, but all had four years of experience with CAD software. The demonstration of SuperSlob, designed and presented by Ryan Tubby, was well received, partly (we believe) because it was conducted as a ‘peer-to-peer’ training session, as opposed to subsequent training in other SuperSlob projects where instruction was ‘external’ (for example, by a student from another year group). Trials of SuperSlob training indicate a need to provide ongoing access to a tutorial resource.

4.3 Influence of Superslob on the Design Process

Observation of Design Island and other current SuperSlob projects has revealed that the plugin has had some interesting influences on the dynamics of the collaborative design process and perceptions of what the models are representing. The first and most dramatic influence is that the introduction of SuperSlob has brought digital modelling towards the beginning of the design process, which has typically occurred only in the later stages of LBM studios. Digital LBM studios have typically programmed four weeks for sketch modeling, six weeks for digital documentation and four weeks for fabrication and assembly. On the surface the early appearance of a digital model would seem an unambiguously positive impact, in that the potential for the student group to proceed to fabrication and assembly is more likely. A SuperSlob model is production-ready. Connections between components have already been incorporated. The visual clarity of the models also lends them to being annotated with a red pen in order to highlight areas of concern and improvement.

It has however been interesting, and slightly concerning, to see how fully slobbed concept models have been interpreted by some students as being ‘resolved’, due to their intricacy and accuracy. The models are considered ‘precious’ and do not invite modification in the same way as cardboard and glue models do. SuperSlob models can be rapidly detailed but there is an assumption as to the invested time and effort.

A noteworthy change in the design dynamic is the way that SuperSlob has been adopted as a tool for ‘collaborative digital projection brainstorming’, where one student runs SuperSlob on a digital projector, building or adapting a model on the basis of live comments from other group members. This process improves most existing mechanisms for group design, promoting the core LBM principle of team involvement in a single session, on one model. The primary advantage of real-time model generation is the short feedback loop and direct communication, resulting in immediate assistance from the team on a multitude of detail decisions.

4.4 Module Calculators

Because the dimension of a SuperSlob module can often be uncommon (for example, 143 mm) the multiples are difficult to calculate quickly. These calculators specifically deal with the relationships between sheet dimensions, module size, and material thickness with the aim of optimising ‘nesting’ efficiency and a simple way to deal with uncommon dimensions. They allow students to enter desired dimensions and select the most suitable module for that particular material thickness, and vice versa.

5. FUTURE DEVELOPMENTS

5.1 Accessibility

Accessibility is important to the Castle for two distinct reasons: to support the design and documentation phases of LBM studios and to introduce a contemporary way of ‘making’ or production to a broader demographic. The time taken by a LBM student group to cycle through model-making phases has been reduced dramatically, meaning that an LBM studio can, once again, expect to fabricate and assemble the designed object. There is however a need to analyse more closely what is happening in the mind of the student designer as the slobbing speed increases and production-ready models proliferate. The accessibility of Superslob to a broader non-designer demographic has yet to be tested. The introduction of contemporary production processes to construction industry trainees is an important component of the mobile factory currently under development. Exercises involving primary school students with Sketchchair have given some preliminary indications of the questions we need to ask in relation to researching the impact of digitally enhanced design. The educational and commercial interest in SuperSlob is raising questions about how the release of the software might be managed, but individual uptake will undoubtedly be affected by the requirement for a SketchUp Pro 8 license.
5.2 Reliability and Predictability
The several million successful slot and tab combinations created by Superslob in the projects described in this paper have demonstrated the reliability and predictability of slobbing alignment. In this sense the software is robust, the designer can focus on the overall component configuration (instead of slobbing alignment) and the assembly of the resulting model is satisfying. However, the rigorous testing undertaken by around sixty students has illuminated several weaknesses of the software including the occasional and apparently random loss of components whilst exporting to nesting software; the memory required to slob a large number of components; the limited number of joint pattern options currently available in the library; the dexterity required to manually edit the slobbing of non-orthogonal objects; and that it is currently possible to create a panel larger than the specified sheet dimension limits. The selected points in future will display a warning if any dimension exceeds the sheet limits.

5.3 Applicability to a Range of Design Situations
Strong interest from both educational and commercial design sectors indicates the construction system and software has potential beyond the initial Castle brief. It is SuperSlob’s placement at the ‘bottom end’ of the parametric options that enables it to build almost any object composed of sheet based components, and allows the possibilities of new ventures for the Castle project such as DI, B1 and F1.

5.4 Other Potential Developments
Communication between designers using Superslob tends to refer to module dimensions, as opposed to conventional unit dimensions. This has led to an aspiration for a live module reading related to the cursor. Other desires include simplification of the steps required to slob an internal shape; the ability to mix material thicknesses within the same file; ‘automatic’ creation of an assembly booklet from the file; the ability to slob ‘off-grid’ (around a radius for example); to have text based references; and increasing the intelligence of the connections so that applying tabs along the edge of a panel could automatically generate complimentary slots on the adjacent panel.

CONCLUSION
The Castle’s innovative single-skin plywood construction system, in addition to the needs of the LBM studios, led to the conception of SuperSlob, while aspirations of closing the gap emerging in digital design between assemblers and designers informed the workflow and choice of the SketchUp platform. While accessibility has yet to be tested beyond UTAS, SuperSlob satisfies three desired characteristics namely, being easy to learn, reliable and predictable, and flexible enough to be applied to a variety of design scenarios. The continual and iterative nature of software development suggests that the results of these early trials with SuperSlob, will lead to several rounds of continual improvement before the eventual release. The interest in SuperSlob that has come from both the educational and commercial ends of the design spectrum is raising questions about how the release might be managed. Currently there is a lean toward releasing SuperSlob as open source or restricted access software.

ACKNOWLEDGEMENTS
Thanks to Trevor Grant, the SketchUp consultant from Liverpool UK, for his expertise in writing plugins and for writing the majority of SuperSlob; Harry Tamms, Rob Kolkert, and the trainees of Youth Futures Inc. for their continual support and collaboration with UTAS on the Castle; thanks to all those who made Design Island so successful; and finally to all those who understood and supported the long days, the late nights, the stress and the tens of thousands of words proof-reading, thank you.

REFERENCES
Diatom Studio, SketchChair® (2012). Retrieved from http://www.sketchchair.cc