

The SKA Cost/Performance Tool: A Hierarchical SKA Modelling Tool

D. Ford¹, R.C. Bolton¹, T. Colegate², P. Alexander¹, and P. Hall²

¹ Cavendish Laboratory, J.J. Thomson Avenue, Cambridge, CB3 0HE, UK

² Curtin Institute of Radio Astronomy, Perth, Western Australia *

Abstract. The SKA Cost/Performance Tool provides a framework in which hierarchical descriptions of telescope designs can be built and costed as a function of input parameters and telescope performance. The tool allows engineers and astronomers to rapidly explore the possible parameter space of SKA designs, probe the cost vs. performance tradeoffs which affect them, and ultimately produce optimised designs for the SKA. In this paper, we give an overview of how the tool works, closing with a brief overview of the graphical user interface used to control it.

1. Introduction

Intrinsic to the design of any telescope are trade-offs between cost and scientific performance (see, e.g., Gaensler and Lazio 2006). In the design of the Square Kilometre Array (SKA), these trade-offs will be unusually complex, for two reasons. Firstly, its design will have an unusually large number of free parameters: there will be many design choices to be made in addition to the familiar considerations of sensitivity and survey speed. For example, it will use a hybrid of detector technologies co-existing side-by-side – both the International SKA Project Office^a (ISPO) Reference Design (ISPO 2006; Schilizzi *et al.* 2007) and the SKADS Benchmark Scenario (Alexander *et al.* 2007; Bolton *et al.* 2009) favour a three-component hybrid of phased aperture arrays, dishes with wide-band feeds and either close-packed aperture arrays or phased-array feeds (PAFs) – and the frequency coverage and collecting area of each component must be decided upon. Within the back-end processing for the aperture arrays, the choice between digital and analogue beam-forming remains to be made; Alexander *et al.* (2007) argue that even if one becomes the favoured option, the other should be developed in parallel for the foreseeable future to provide a realistic fallback alternative. Within the dish-based collectors, issues such as the dish size and feed type remain to be finalised.

The second source of complexity in the SKA's cost-performance trade-offs is the multi-faceted nature of its science programme (see, e.g., Gaensler 2004; Jones 2004). Its prospective users each place different requirements upon the SKA's angular resolution, survey speed, sensitivity and frequency coverage, and a compromise will have to be reached (see, e.g., Jackson 2003, 2006).

Whilst the ISPO Reference Design and the SKADS Benchmark Scenario documents have presented and discussed

in detail a range of manually-optimised skeletal designs for the SKA, it is clear that more immediate access to cost estimates and scientific simulations of telescope designs will be essential before engineers can make significant progress in charting the available parameter space. To this end, the SKA Cost/Performance Tool (Ford *et al.* 2009) has been developed on behalf of the SKA Program Development Office (SPDO) and in part as work package DS3-T3 of the SKADS programme. This tool acts upon telescope designs which are described in a hierarchical fashion, in which large *design blocks* – for example, the whole SKA – subdivide into smaller units – for example, SKA Stations – until eventually the hierarchy reaches *components* – indivisible atomic units. One possible view of how this hierarchy might appear is shown in Figure 1. The tool can calculate the number of components required to build any given design block, propagate costs through this hierarchy to sum up to total cost of any given telescope design, and provide a breakdown of the contributions of each design block to the total cost. In addition, the tool can also propagate other arbitrary quantities through the hierarchy, including, for example, the power consumption of components and their data rates. The former is useful for calculating the total power requirements of the SKA, meanwhile the latter is an unusual case, where data processing steps can *reduce* the total data flow to their parents.

A key feature of the Cost/Performance Tool is that the hierarchical telescope designs which it acts upon are *scalable*; they do not represent a single telescope built to a particular specification, but rather take a wide range of input parameters, and within reasonable limits, can estimate what components would be required to build a telescope to any requested specification. This does not replace the need for specialist engineers: the calculations performed by the tool are *estimates* based upon simple models, which would need to be extensively expanded upon before they could actually be built. In the first instance, our aim is to produce realistic models of how the cost of the SKA will *scale* with design parameters, not to produce accurate absolute costings.

The Cost/Performance Tool builds upon the work of its predecessor, *SKAcost*, which was developed by the Australia

* This work was supported by the European Commission Framework Program 6, Project SKADS, Square Kilometre Array Design Studies (SKADS), contract no 011938.

^a The International SKA Project Office is now the SKA Program Development Office (SPDO).

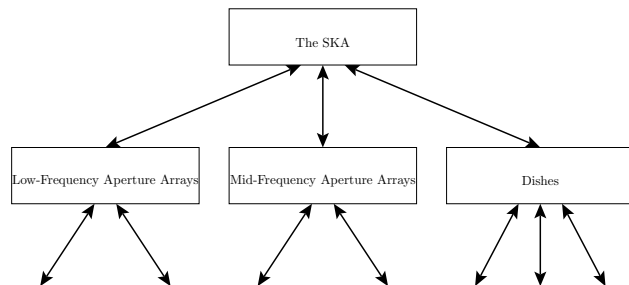


Fig. 1: A simplified example of how the top of a hierarchical telescope design might appear as a network of design blocks; this example is based loosely upon the structure of the SKADS Benchmark Scenario (Alexander *et al.* 2007).

Telescope National Facility (ATNF), with some financial support and system engineering input from the ISPO. *SKAcost*, which itself used the modular approach first described in Horiuchi *et al.* (2004), was also a hierarchical tool which could estimate component costs as a function of purchase date for dish-based SKA designs (Chippendale *et al.* 2007; Schilizzi *et al.* 2007).

2. The Structure of the Cost/Performance Tool

Within the Cost/Performance Tool, we maintain a sharp division between the *costing engine* – the software used to calculate the numbers of components needed to build an SKA to a particular specification – and the *telescope designs* which it acts upon. This separation ensures that telescope designs can be modified by engineers without exposure to the lower-level software implementation of the cost calculations; it also ensures that cost calculations are performed in a homogeneous way between all design blocks, since they must all use the same costing routines. In addition, the costing engine is separated from the user interfaces which it uses to communicate with the user and with the outside world. This structure is illustrated in Figure 2.

Three user interfaces to the Cost/Performance Tool are currently available. The majority of users use the graphical interface on account of its user-friendliness. Some users with scripting experience prefer the commandline interface, which provides a similar degree of flexibility to the graphical interface. Finally, more experienced programmers can use the Cost/Performance Tool directly from within the Python programming language, where they are able to develop their own independent user interfaces to the tool, specialised for their own particular requirements. In view of the sophistication of the new tool, there is currently no web-based interface to match that which was previously available for *SKAcost*, and we envisage that users will run the graphical user interface on their own local machines.^b

As Figure 2 indicates, the Cost/Performance Tool can also be accessed by other applications via socket-based inter-

^b The Cost/Performance Tool is known to run under Microsoft Windows, Mac OS X and Linux. It is believed to run under all other POSIX-compliant operating systems.

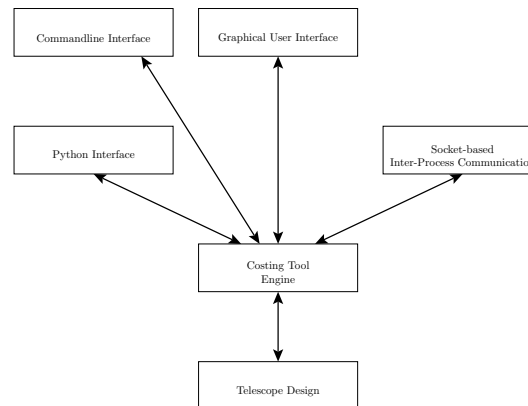


Fig. 2: A diagram of the modular structure of the Cost/Performance Tool. The costing engine is separated both from the telescope design models it acts upon, and from the interfaces with which it communicates with the user and with other software packages. This means that the user is able to modify telescope designs without detailed knowledge of how the costing engine works, and that experienced programmers can use the Cost/Performance Tool from within the programming language of their choice.

process communication, which means that other design optimisation tools can query live cost estimates for designs from the tool. As a proof of this concept, a SKADS tool is under development which will optimise the cable and trenching layouts between SKA stations. It will communicate with the Cost/Performance Tool to obtain information about the fibre cost model, and will communicate results back into the cost calculations.

As well as the division which we draw between telescope designs, the Cost/Performance Tool engine and its user interfaces, we also divide up telescope designs into separate sharply-divided blocks. Each design block within a telescope design hierarchy is a *black box*: it takes a defined set of inputs from its *parent* design blocks, operates on them in some way which other design blocks need not be concerned with, and then returns a defined set of outputs. This information flow is illustrated in Figure 3. This approach means that the various design blocks in a telescope design can be efficiently written by a variety of different engineers working at different institutions, each bringing their own specialist expertise to their own particular areas. Because each block is a black box, each can straightforwardly be re-implemented in the light of new information or expertise, without changing the rest of the telescope design around it. Within the Cost/Performance Tool, it is straightforward to swap one design block for another within a telescope design, to obtain a rapid comparison of the costs calculated by different models.

In addition, the Cost/Performance Tool makes it easy to re-use components and design blocks in many different places in a hierarchical telescope design if they are designed to take a sufficiently general set of inputs that they can be used in several different situations.

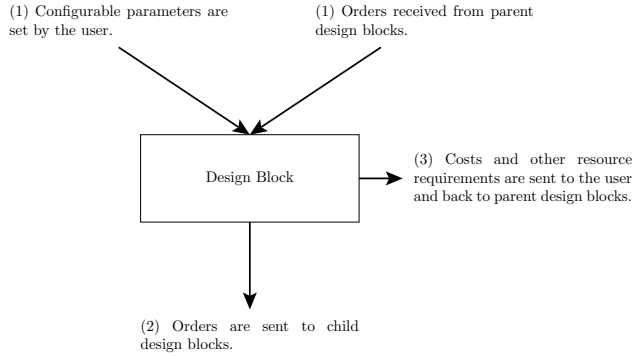


Fig. 3: A diagram of the data flow which takes place within design blocks. First of all, in Step (1), the design block receives *orders* from all of the parent design blocks which make use of it, together with some user-configurable parameters. The design block then, in Step (2), computes how many child design blocks are needed to fulfill these orders, sends orders through to child design blocks, and calculates any additional “glue” costs which are associated with putting these sub-components together. Finally, the design block returns its total resource requirements – not just cost, but also power consumption and any other arbitrary attributes which the user wishes to propagate through the hierarchy – to the user and to its parent design blocks.

3. The Graphical User Interface

In Figures 4, 5, 6 and 7, we show four views of the Cost/Performance Tool’s graphical user interface (GUI). To demonstrate its use in practice, it is necessary to have a telescope design for it to act upon. The process of putting together complete and realistic telescope designs in the tool will require extensive consultation with expert engineers throughout the SKA community, which we are currently actively engaged in. In the meantime, we have developed an *Example Telescope*, which is loosely based upon the Benchmark Scenario from the Second SKADS Design and Costing Memo (Bolton *et al.* 2009). This design is not intended to serve any purpose other than as an example, and is not presented as a fully-costed SKA design. To emphasise this, we have chosen to display all design block costs in a fictitious currency which we have called the SKA Accounting Unit (SAU); its value roughly matches the cost of the whole SKA.

For reference, the Example Telescope consists of 250 mid-frequency aperture array stations, each 24 m in radius, 250 low-frequency aperture array stations, each 82 m in radius, and 2,480 dishes, each 15 m in diameter. It lacks any model of the correlation or computational hardware, for which arbitrary placeholder costs are used.

4. Summary

We have constructed a tool which allows the user to construct hierarchical telescope designs in a graphical environment, and which can then propagate costs, power consumption, data rates, and other arbitrary quantities through these hierarchies to evaluate the total cost of any given telescope. We and the SKA Program Development Office (SPDO) are working in active collaboration with specialist engineers to produce a range of

Name	Number Built	Intrinsic Cost	Cumulative Cost	Block Type
SKA Dish_AA	0	0 SAU	888.54e+3 +/- 1E.00e-3 SAU (NPV in 2009)	Design Block
Example_Comms_Dish_Link	18.16e3	0 SAU	231.25e+3 SAU (NPV in 2009)	Design Block
AAIn_Core	0	0 SAU	289.27e+3 +/- 6.55e-3 SAU (NPV in 2009)	Design Block
Dish_Small_Parabolic_Parametric	2.08e3	0 SAU	262.57e+3 SAU (NPV in 2009)	Design Block
Dish_Small_Parabolic_Parametric_ScalingFactor	1.85e3	0 SAU	262.57e+3 SAU (NPV in 2009)	Component
Example_Comms_Dish_Link	2.48e3	0 SAU	175.54e+3 SAU (NPV in 2009)	Design Block
Dishes_Core	0	0 SAU	171.89e+3 SAU (NPV in 2009)	Design Block
Dishes_Core	0	0 SAU	171.74e+3 SAU (NPV in 2009)	Design Block
Example_AA_Station_Link	2.87e3	0 SAU	168.75e+3 SAU (NPV in 2009)	Design Block
AAIn_Core	0	0 SAU	167.25e+3 SAU (NPV in 2009)	Design Block
Example_Comms_Mid_Shape_Array_Block_10G_10G	21.36e3	0 SAU	114.46e+3 SAU (NPV in 2009)	Design Block
Comms_10G_Mid_Shape_Array_DFA	21.36e3	107.28e+3 SAU (NPV in 2009)	107.28e+3 SAU (NPV in 2009)	Design Block
AA_Order	0	0 SAU	100.70e+3 +/- 5.20e-3 SAU (NPV in 2009)	Design Block
AAIn_Station	250	0 SAU	87.75e+3 +/- 8.54e-3 SAU (NPV in 2009)	Design Block
AAIn_Station	2.34e6	0 SAU	6.408e+3 +/- 9.52e-3 SAU (NPV in 2009)	Design Block
AAIn_Station_Infrastructure	250	0 SAU	60.84e+3 +/- 5.21e-3 SAU (NPV in 2009)	Design Block
Example_Comms_Prv_Amplification_Block_10G_10G	15.08e3	0 SAU	54.56e+3 SAU (NPV in 2009)	Design Block
Comms_10G_10GFlow_Code	23.66e3	0 SAU	45.13e+3 SAU (NPV in 2009)	Component
Comms_10G_Protocol_DFA	15.08e3	48.43e+3 SAU (NPV in 2009)	48.43e+3 SAU (NPV in 2009)	Design Block
AAIn_Station_Material	130.86e3	47.42e+3 +/- 8.52e-3 SAU (NPV in 2009)	47.42e+3 SAU (NPV in 2009)	Design Block
Example_Comms_Signal_Generation_Receiving_Tx	798.83e3	0 SAU	40.81e+3 SAU (NPV in 2009)	Design Block
Example_Dish_Order_Comms	798.83e3	0 SAU	40.82e+3 SAU (NPV in 2009)	Design Block
AAIn_Station_Cabling	5.25e6	0 SAU	39.89e+3 +/- 5.73e-3 SAU (NPV in 2009)	Design Block
AAIn_Station	18.46e6	0 SAU	38.29e+3 +/- 6.28e-3 SAU (NPV in 2009)	Design Block
Laborer_Field	138.28	33.15e+3 SAU (NPV in 2009)	33.15e+3 SAU (NPV in 2009)	Component
AA_Station_Processor_Unit_Try	4.04e6	26.06e+3 +/- 5.20e-3 SAU (NPV in 2009)	26.06e+3 +/- 5.20e-3 SAU (NPV in 2009)	Design Block
Comms_10G_700MHz_Cable	22.27e3	24.68e+3 SAU (NPV in 2009)	24.68e+3 SAU (NPV in 2009)	Component
Comms_10G_Linear_Filter	308.57e3	24.62e+3 SAU (NPV in 2009)	24.62e+3 SAU (NPV in 2009)	Component
CAT7_Cable	75.21e3	22.58e+3 +/- 6.83e-3 SAU (NPV in 2009)	22.58e+3 +/- 6.83e-3 SAU (NPV in 2009)	Component
Example_Comms_Signal_Generation_Receiving_Tx	798.83e3	0 SAU	23.89e+3 SAU (NPV in 2009)	Design Block
Comms_10G_Laser_Bias	18.64e3	26.14e+3 SAU (NPV in 2009)	26.14e+3 SAU (NPV in 2009)	Design Block
AA_Order	0	0 SAU	18.29e+3 +/- 1.63e-3 SAU (NPV in 2009)	Design Block
Comms_10G_Receiver	348.98e3	19.27e+3 SAU (NPV in 2009)	19.27e+3 SAU (NPV in 2009)	Design Block
AA_Order	0	0 SAU	14.82e+3 SAU (NPV in 2009)	Design Block
Infrastructure_Cable	820e3	14.82e+3 SAU (NPV in 2009)	14.82e+3 SAU (NPV in 2009)	Component

Fig. 4: A spreadsheet of all of the components and design blocks in the Example Telescope Design, as seen in the tool’s graphical user interface. In this figure, we have chosen to sort the blocks in order of their cumulative cost, so that the most expensive block, the whole telescope, appears at the top. Further details are in the text.

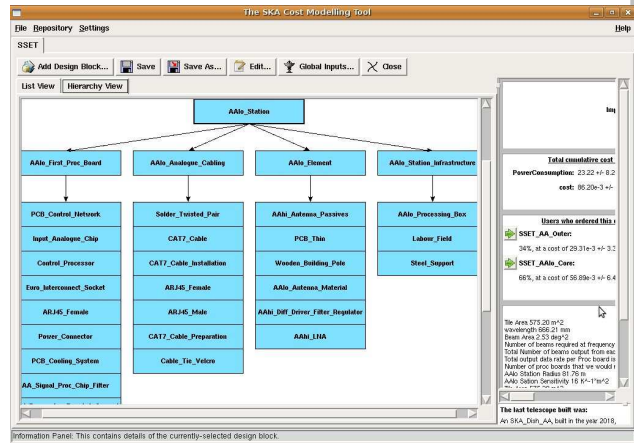


Fig. 5: The hierarchy of design blocks which are required to build an *AAIn_Station*, as seen in the tool’s graphical user interface. Where strings of blocks appear in vertical lists, all of the blocks in the list are children of a common parent above, but they are too numerous to be displayed horizontally.

reference telescope designs within this framework. Over the course of the PrepSKA programme, we also plan to link the Cost/Performance Tool into technical simulations of the SKA’s science performance.

Acknowledgements. This work is supported by the European Community Framework Programme 6, Square Kilometre Array Design Studies (SKADS), contract number 011938.

References

Alexander P., Faulkner A., Torchinsky S., van Ardenne A., Wilkinson P., de Vos M., Bakker L., Garrington S., Harris G., Ikin T., Jones M., Kant D., McCool R., Patel P., *SKADS*

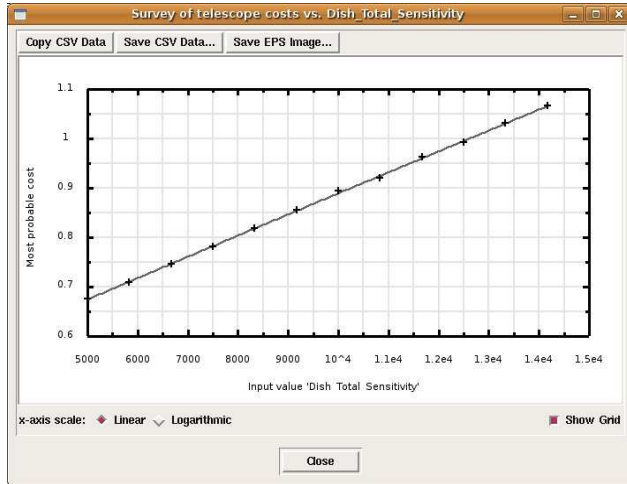


Fig. 6: A parameter survey of the cost of the SKA in SAU as a function of the desired sensitivity of the dish component, measured along the horizontal axis in m^2/K . This demonstrates the power of the scalable telescope designs used by the tool; rather than providing a single cost estimate for a single telescope, they provide an indication of how cost scales with performance.

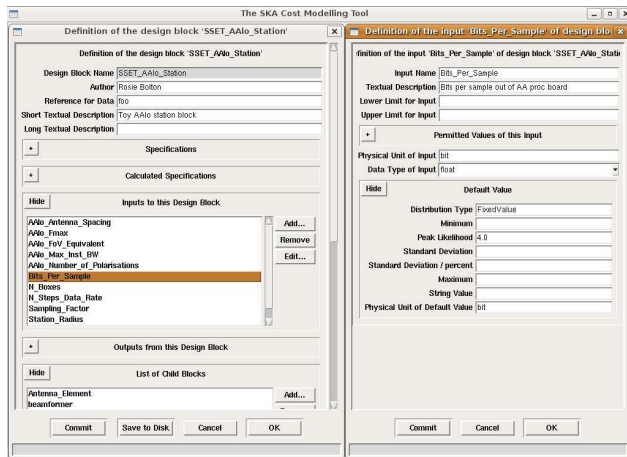


Fig. 7: The Cost/Performance Tool's graphical design block editor, here editing the *AAlo_Station* design block. The left panel shows the design block's definition, including a list of its inputs. In the right panel, we have opened the *Bits_Per_Sample* input, to show that it is measured in physical units of bits and that its default value is 4 bits per sample.

Benchmark Scenario - Design and Costing, SKA Memo 93, 2007

Bolton R., Faulkner A., Alexander P., Torchinsky S., van Ardenne A., Wilkinson P., de Vos M., Bakker L., Garrington S., Harris G., Ikin T., Jones M., Kant D., Kettle D., McCool R., Patel P., Romein J., *SKADS Benchmark Scenario Design and Costing - 2*, SKA Memo 111, 2009

Chippendale A. P., Colegate T. M., O'Sullivan J. P., *SKAcost: a Tool for SKA Cost and Performance Estimation*, SKA Memo 92, 2007

Ford D. C., Bolton R. C., Colegate T. M., Alexander P., Hall P.,

The SKA Costing and Design Tool, SKADS Technical Memo T23, submitted as an SKA memo, 2009

Gaensler B., *Key Science Projects for the SKA*, SKA Memo 44, 2004

Gaensler B., Lazio J., *Trade-offs Between Science and Engineering*, SKA Memo 82, 2006

Horiuchi S., Chippendale A., Hall P., *SKA system definition and costing: a first approach*, SKA Memo 57, 2004

ISPO, *Reference Design for the SKA*, SKA Memo 69, 2006

Jackson C., *SKA Science: A Parameter Space Analysis*, SKA Memo 29, 2003

Jackson C., *SKA Key Science Requirements Matrix 2006; Prime Science Drivers*, SKA Memo 83, 2006

Jones D. L., *SKA Science Requirements*, SKA Memo 45, 2004

Schilizzi R. T., Alexander P., Cordes J. M., Dewdney P. E., Ekers R. D., Faulkner A. J., Gaensler B. M., Hall P. J., Jonas J. L., Kellermann K. I., *Preliminary Specifications for the Square Kilometre Array*, SKA Memo 100, 2007