Task Description

Title	Senior Scientific Software Engineer
Education	University degree in Computer Science, Physics, Electrical Engineering or equivalent
Work Description	The work involves two key areas 1. Developing data reduction pipelines for the operational instruments at astronomical observatories. The data reduction pipelines are used in two operational environments; by Data Flow Operations at Munich; and by the Science Operations; for the quick-look assessment of data; the health-monitoring of instruments and detectors; the generation of master calibration data; the reduction of scientific exposures; and data quality control. The work involves the preparation of data reduction procedures for the pipelines, and the implementation of these parts in ANSI-C and the High-Level Data Reduction Library (HDRL). 2. The Exposure Time Calculators are a suite of Web applications provided for the Call for Proposals. The total usage of the system peaks to about 35 000 simulation runs during the Call for Proposals periods in March and September each year. The work involves maintenance and development of C++ and Python ETC Web applications for the Interferometers. The majority of the work involves the first activity.
Experience and Knowledge	He or she is a software engineer, ideally, with … The following mandatory skills/experiences • minimum 3 years experience in developing scientific data processing applications in ANSI-C and C++ • good knowledge of processing and statistical handling of scientific data • experience in a Unix/Linux environment. • able to participate in any of the maintenance and development efforts contributing to the entire software life-cycle, from analysis to design, coding, documentation and testing, to deployment and maintenance. • experience in software development tools (version control, development environments, debuggers, profilers) • ability to interact with the instrument scientists at the operational sites for the specification of data reduction procedures. • ability to collaborate with the team; take responsibility for the assigned activities and meet the agreed deadlines. • excellent command of the English language, very good communication skills, The following desireable skills/experiences • Excellent knowledge of image and signal processing methods • Excellent knowledge of Web technologies and frameworks • Good knowledge of the Python programming language • Knowledge of astronomical instrumentation and observational techniques in astrophysics (imaging, adaptive optics, spectroscopy) and the related data reduction methods

	• Experience with multi-threaded programming
Location	Munich, Germany
Start Date 0	01.01.2015
Duration	36 months and option to extend by 24 months – Full time.
Project Milestones	The list below details the tasks to be carried out during the first 2 years
01-03-2015	**Task #1 – High-Level Data Reduction Library: Strehl Ratio Calculation** **Purpose** The Strehl is a measure of the quality of an Adaptive Optics correction system on the telescope. It is defined as the ratio of the peak image intensity from an observed point source compared to the maximum attainable intensity using an ideal optical system limited only by diffraction over the telescope aperture. The Strehl algorithm involves the following data reduction steps: • Determination of the peak of the observed Point Spread Function (PSF) peak • Determination of the ideal PSF peak • Computation of the ratio between the two measured PSF peak values and the associated error A common algorithm for the computation of the Strehl ratio will be developed for the High-Level Data ReductionLibrary. **Description** • Review the specifications described in the Strehl project document. • Implement the HDRL functions and a recipe using the Common Pipeline Library and HDRL (ANSI-C) • Compare the implementation with results obtained from an IDL program. • Prepare user documentation; implement unit tests and NRI regression tests • Provide support to the pipeline developers for the upgrade of existing pipelines
01-06-2015	**Task #2: High-Level Data Reduction Library: Source extraction** **Purpose** Finding and extracting fundamental parameters of astronomical sources (e.g. position, flux, background, ellipticity) is a major task in astronomical data analysis. The Source extraction algorithm involves the following data reduction steps: • Find all the pixels belonging to an objects/source in the image by analyzing the light distribution • Calculate fundamental source parameters based on the light distribution of the pixels (e.g. barycenter, total flux) • Repeat this for all the objects of an image and create a catalogue of fundamental parameters **Description** • Study the interface of the source extraction module prototype and adapt it for the usage in HDRL

	• Integrate the module in the HDRL library following HDRL standards • Prepare user documentation; implement unit tests and NRI regression tests • Provide support to the pipeline developers for the upgrade of existing pipelines
01-10-2015	**Task #3 – High-Level Data Reduction Library: Optimal Extraction** **Purpose** Extracting the Spectral Energy Distribution (SED) of an object observed in a spectrograph should be done by optimizing the signal-to-noise ratio. If the shape of the signal is known, various algorithms described in the literature (e.g. Horne) are available to achieve this goal. Common algorithms for the optimal extraction of spectral energy distributions will be developed for the High-Level Data Reduction Library. Work will be based on the adaptation and generalization of algorithms available in the literature and the ones used in the pipelines. **Description** • Review the specifications with the quality control and instrument scientists • Collect a set of test data for the validation of the results. • Implement the HDRL functions and a recipe using the Common Pipeline Library and HDRL (ANSI-C) • Prepare user documentation; implement unit tests and NRI regression tests • Provide support to the pipeline developers for the upgrade of existing pipelines
15-02-2016	**Task #4 – High-Level Data Reduction Library: Fringe Correction** **Purpose** Fringes are interference patterns caused by multiple internal reflections within unevenly thinned charge-coupled devices (CCD). Owing to the specific thickness of visual-band CCD's, fringing will occur in the red end of the detector sensitivity scale (i.e. most relevant in the i and z bands). Fringing is apparent in virtually all instruments operating in the visual wavelength range, but is probably most apparent in the UV-sensitized detectors. A common algorithm for the extraction and removal will be developed for the High-Level Data Reduction Library. Work will be based on the adaptation and generalization of an Austrian algorithm. **Description** • Review the specifications described in the Austrian project • Collect a set of test data for the validation of the results. • Implement the HDRL functions and a recipe using the Common Pipeline Library and HDRL (ANSI-C) • Prepare user documentation; implement unit tests and NRI regression tests • Provide support to the pipeline developers for the upgrade of existing pipelines

01-04-2016	**Task #5 – Adaptation of Pipelines to the Mosaicing Reflex Workflow** **Purpose** An instrument independent image stacking and mosaicing workflow based on the -Reflex engine and CPL pipeline recipes is available, making use of the Terapix image combination tools, and supports optical and infrared instruments. The workflow is instrument-generic and takes input from all imaging pipelines. The task consists of finalizing the package in view of preparing the public release version of this workflow. **Description** • Collect additional test data and review the science-grade workflow requirements. • Enhance the workflow in collaboration with the instrument scientists and pipeline developers • Help the pipeline developer to produce all the pipeline products to run the workflow • Enhance and finalise the workflow user manual and tutorial documentation • Prepare regression tests, installation procedures, and workflow release
01-10-2016	**Task #6 – Enhancement of the ETC Calculation Framework** **Purpose** The Exposure Time Calculators (ETCs) are a suite of Web applications provided for the Call for Proposals. The ETCs provide tools to astronomers to predict the signal to noise ratio - or conversely, the exposure time to achieve it - under a set of assumptions about the configuration and performance of an instrument and the observing conditions at the observatories. The total usage of the system peaks to about 35 000 runs during the Call for Proposals periods in March and September each year. More than twenty ETCs for instruments are currently provided, as well as tools for the VLTI interferometry facilities. The HTML/JavaScript based user interface on the client side interacts through a CGI interface with a back-end calculation engine framework on an Apache web server running in a Linux environment. The back-end system is implemented in C++ and scripting languages including Python and shell scripts. The work involves participation in a process of defining and constructing a fully or partially new implementation of the ETC system, in particular the back-end calculation engine, as well as the conversion and migration to a new framework. **Description** • The work includes the following activities, in a small team of developers. • Study and participate in maintenance and development the existing ETC system • Collect and define requirements for the new ETC framework with scientists and developers

	• Study and evaluate technology options for an ETC calculation engine rebuild • Build and evaluate prototypes • Build the new ETC calculation framework based on the outcome of the evaluation • Write documentation for maintainers and users • Prepare unit and regression tests as well as installation procedures
15-12-2016	**Task #7– Migration of existing ETCs to the new framework** **Purpose** In continuation of task #6, the existing ETC applications, shall be converted and migrated to the new system; documented, tested, validated and put into operation. The core of the current back-end system is a set of C++ applications and a libraries of auxiliary models and utility functions. The applications work with collections of telescope- and instrument specific calibration data files, instrument definition files, various astronomical data template files, a dictionary structure of input- and output parameters, as well as HTML templates which are combined by a C++ module to collect and present the numeric and graphical results of the calculations in the client browser. It must be carefully ensured and documented that the instrument performances predicted with the new ETC implementations are consistent with the current ones. **Description** • Convert the instrument specific instrument definition files to the new framework • Convert the calibration data files to the new framework • Implement all individual ETC models fully to the new framework • Test and validate the re-implemented ETCs in cooperation with the scientists and ETC development team • Prepare documentation • Evaluate options for tools for automated testing tools • Define and build a regression test suite