OPTIMISING MEASUREMENT PROCESSES USING AUTOMATED PLANNING

S. Parkinson* and A. Crampton and A. P. Longstaff

Department of Informatics, University of Huddersfield,
HD1 3DH, UK
**E-mail: s.parkinson@hud.ac.uk*

Many commercial measurement processes are planned with little or no regard to optimality in terms of measurement time and the estimated uncertainty of measurement. This can be because the complexity of the planning problem makes optimality in a dynamic environment difficult to achieve, even with expert knowledge. This paper presents a novel approach to measurement planning using automated planning. Detailed information regarding the modelling and encoding of measurement processes are provided. The benefits of this approach are demonstrated through the results of applying it to machine tool calibration. A discussion is then formed around the development of future tools to further validate the approach.

1. Introduction

Measurement processes often contain multiple measurements, which have time and order dependencies when estimating and minimising the uncertainty of measurement. The scheduling of interrelated measurements can have significant impact on the estimated uncertainty of measurement, especially in dynamic environments such as those taken within non-stable environmental temperature. Expert knowledge is required to produce both valid and optimal measurement plans. This can present problems for industrialists who are wanting to implement or improve their measurement processes.

The Guide to the expression of Uncertainty in Measurement (GUM)[?] establishes general rules for evaluating and expressing the uncertainty of measurement with the intention of being applicable to a broad range of measurements. An expert will use the GUM to plan and optimise a sequence of measurements by making informed decisions. However, it is often the case that planning a sequence of measurements against a continuously changing environment (e.g. changing temperature) can be cumbersome and little or no regard is taken to optimality, resulting in a higher uncertainty than that which is achievable.

The GUM, and other theoretical guides, contain detailed, advanced procedures for estimating the uncertainty of measurement. However, this can often make it difficult to implement on an industrial level. The Procedure for Uncertainty MAnagement (PUMA)[?] provides an iterative method for reducing the estimated uncertainty per measurement. However, this approach does not consider scheduling a sequence of measurements to reduce the overall estimated uncertainty of measurement.

This paper proposes an approach that utilises Automated Planning (AP) to encode and deliberate over the measurements, optimising their order by anticipating their expected outcome. The theory of automated planning and the implementation of knowledge are discussed in the following two sections. This leads to a demonstration of how expert knowledge can be encoded and subsequently used to produce optimal measurement plans. A discussion is then formed around the authors' ambition to develop and extend tools which will enable users to easily optimise their measurement processes.

2. Automated Planning

Planning is an abstract, explicit deliberation process that chooses and organises actions by anticipating their expected outcome. Automated planning is a branch of Artificial Intelligence (AI) that studies this deliberation process computationally and aims to provide tools that can be used to solve real-world planning problems[?]. To explain the basic concepts of AP, a conceptual model is provided based on the state-transition system. A state-transition system is a triple $\sum = (S, A, \rightarrow)$ where $S = (s_1, s_2, \dots)$ is a finite set of states, $A = (a_1, a_2, \dots)$ is a finite set of actions, and $\rightarrow: S \times A \rightarrow 2^s$ is a state-transition function. A classical planning problem for a restricted state-transition system is defined as a triple $P = (\sum, s_0, g)$, where s_0 is the initial state and g is the set of goal states. A solution P is a sequence of actions $(a_1, a_2, \dots, a_k)$ corresponding to a sequence of state transitions $(s_1, s_2, \dots, s_k)$ such that $s_1 = \rightarrow (s_0, a_1), \dots, s_k = \rightarrow (s_{k-1}, a_k)$, and s_k is the goal state. The state s_1 is achieved by applying action a_1 in state s_0 and so on.

In AI planning, when planning for a complex problem, it can become practically impossible to represent explicitly the entire state space; since the number of states can potentially increase exponentially. In classical planning, the state of the world is represented by a set of first-order predicates which are set true or false by an action a. An action has three elements: (1) a parameter list that is used for identifying the action, (2) a list of preconditions precond(a) that must be satisfied before the action can be executed, and (3) an effect effects(a) that contains a list of predicates that represent the resulting state from the execution of this action.

A full conceptual model for planning is shown in Figure 1 (Modified from[?]). The model has three parts: (1) a planner, (2) a controller, and (3) the state-transition system. The planner generates a plan (sequence of actions) for a specified problem model by using the domain model. A domain model is an abstraction of the real-world domain which is sufficient to be used in conjunction with a planner to automatically solve the planning problem specified in the problem model. A planning problem consists of an initial and goal state composed of a set of first-order predicates. A controller observes the current state of the system from the state-transition function and chooses an action that is generated by the planner based on the domain model. The state-transition system progresses according to the actions that it receives from the controller.

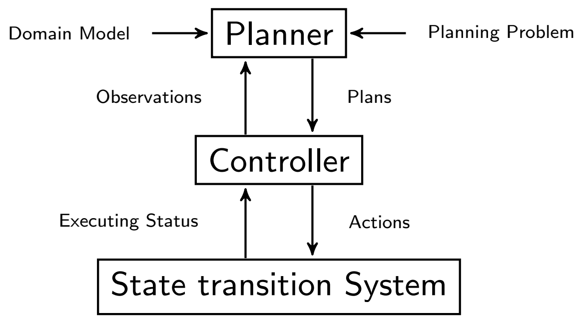

Fig. 1. A conceptual model of AI planning.

3. Knowledge Engineering

Knowledge Engineering (KE), for automated planning, is the process that deals with acquisition, formulation, validation and maintenance of planning knowledge; where the key product is the domain model. To enable a wide use of planning applications, the Planning Domain Definition Language (PDDL)[?] is used to encode the domain. A PDDL problem is comprised of two parts. Firstly, a domain that consists of predicates and actions, and secondly the problem definition, consisting of the initial and goal state.

Domain engineers will typically either develop domain models using (1) a traditional text editor, or (2) a Graphical User Interface (GUI). Traditionally, all domain models had to be developed in a text editor (e.g. Notepad), but recent improvements in GUI knowledge engineering tools are helping to make knowledge engineering a more efficient process. One of the more prominent tools available for domain engineering is itSIMPLE[?] which provides an environment that enables knowledge engineers to model a planning domain using the Unified Modelling Language (UML) Standards[?]. This is significant as it opens up the potential use of the tool to most software engineers with knowledge of UML, but not necessarily AP. itSIMPLE, just like many other tools, focuses on the initial phase of a disciplined life-cycle, facilitating the transition of requirements to formal specification. The design life-cycle goes from gathering requirements and modelling them in UML, right through to the generation of a PDDL model which can be used with state-of-the-art planning tools. The current state-of-the-art in knowledge engineering for AP is sufficient for initial development of real-world applications. However, as the domain advances, features that are not supported by knowledge engineering tools are required. Therefore, for the application presented in this paper, a traditional text editor is used.

3.1. *Implementation of Measurement Knowledge*

In this section, the knowledge required to automatically construct measurement plans, as well as the methods of encoding it, are presented and discussed.

3.1.1. *Temporal Information*

Within metrology, especially industrial metrology, the financial cost of a measurement process can be related to the time it takes to complete. The direct labour cost and any lost revenue due to 'opportunity cost' if measuring a production asset. For modelling purposes, each individual measurement can be broken up into individual temporal components. For example, when performing a measurement, equipment will need to be set-up, the measurement will be performed, and then the equipment will be removed. To enable planning with time, the durative action model of PDDL2.1[?] is used.

In PDDL, a durative action encoding includes a numeric fluent which represents a non-binary resource and can be used in the duration, preconditions and effects of an action. The effects use operators (`scale up, scale down, increase, decrease and assign`) to modify the value of the fluent by using the binary functions (`+, -, /,*`). Comparisons between fluents is performed by using comparators ($\leq$,$<$,$=$,$>$,$\geq$) between functions or fluents and real numbers. Durations are expressed either as a predetermined value, or dynamically using binary functions. For example, the following PDDL syntax for the set-up action `:duration(= ?duration (setup-time ?in ?mv))` specifies that a chosen action will take a quantity of time specified in the initial state for when the instrument `?in` is chosen to take measurement `?mv`.

3.1.2. *Uncertainty Contributors*

Factors that contribute to the total uncertainty of measurement are also encoded as numeric fluents which are specified in the initial state and expressed through action effects. For example, Equation 1 can be easily encoded in PDDL, as provided in Figure 2. Equation 1 shows how to estimate the uncertainty contribution when using a laser interferometer, where $U_{\text{calibration}}$ is provided on a device's calibration certificate. Here L is the length in metres and k is the coverage factor

$$u_{\text{device laser}} = \frac{U_{\text{calibration}} \times L}{k} \tag{1}$$

```
(*(/(k_value ?in)(*(u_calib ?in)(length-to-measure ?ax ?er)))
(/(k_value ?in)(*(u_calib ?in)(length-to-measure ?ax ?er))))
```

Fig. 2. Example PDDL uncertainty effect.

3.1.3. *Dynamics*

Throughout the measurement process, dynamics such as the continuous change in temperature, affect the estimated uncertainty. In order to optimise the measurement process effectively, it is important that such dynamics are encoded into the model. In PDDL, dynamics in the measurement process are encoded either using PDDL2.1 or PDDL+. In PDDL2.1, dynamics can be represented as effects of continuous change throughout an action's duration. For example, `(increase (temperature ?t) (*`

`#t (rate-of-change ?r)))` describes how the environment temperature, `?t`, increases continuously, as a function of the rate-of-change of `?r`. In PDDL+, numerics of continuous, non-linear change can be implemented using the `stop, start process` model exhibited through `processes` and `effects`[?] . However, there is currently no planning tool capable of supporting the full PDDL+ syntax. The solution is to discretise the continuous change into a set of durative actions with time-dependent continuous effects. However, this requires pre-processing of non-linear resources to discretise them based on a discretisation threshold. If the chosen value is too low, then too many actions could be generated rendering the planner unable to solve the problem. If the value it too high, the discretisation could no longer be representative and lead to the generation of suboptimal plans.

3.1.4. *Optimisation*

Based on ISO recommendations, the root of the sum of squares is used to calculate the combined uncertainty[?] . The square root function is not a PDDL operator. However, Considering that the square root is a monotonic function, minimising the sum of the squares is as optimal as minimising the square root of the sum of the squares. In the PDDL model this can be achieved by combining the individual, squared contributions for each measurement, and then adding this to an accumulative uncertainty value, U. This optimisation metric is encoded to minimise the global uncertainty value. For applications where time to measure is cost sensitive, it is possible to minimise the total measurement time, T. It is also possible to perform multi-optimisation by calculating the arithmetic mean of both T and U. However, this could be expanded to the weighted optimisation
$\alpha U + (1 - \alpha)T$, $0 \leq \alpha \leq 1$.

4. Example: Machine Tool Calibration

In current work, AP has been successfully applied to the calibration of precision machine tools where multiple measurements are performed to determine the machine's accuracy and repeatability[?, ?] . This has been achieved by encoding the planning problem in the PDDL2.1 planning language[?] alongside a state-of-the-art planning tool (LPG-td[?]).

Empirical analysis has shown that it is possible to achieve a reduction in machine tool downtime greater than 10 % (12:30 to 11:18 (hh:mm)) over expert generated plans. In addition, the estimated uncertainty due to the schedule of the plan can be reduced by 59 % (48 µm to 20 µm)[?] .

Further experiments have investigated the trade-off when optimising calibration plans for both time and the uncertainty of measurement. We have demonstrated that it is possible to optimise functions of both metrics reaching a compromise that is on average only 5 % worse than the best-known solution for each individual metric[?] . Additional experiments, using a High Performance Computing architecture, show that on average, optimality of calibration plans can be further improved by 4 %. This gain was due to the planner having access to more powerful hardware and so could explore more plans in a reduced time. However, the 4 % improvement demonstrates that in most cases it is sufficient to use a standard PC architecture.

5. Conclusion and Future Challenges

The successful application in the machine tool calibration domain has highlighted the possibility to extend and generalise the technology for a wide variety of measurement problems. The diversity of the problems means that there is no single planner that can be used for all. For example, some planning problems are rich in constraints restricting the measurements, whereas some are rich in temporal and numeric information. In the Automated Planning (AP) community, planners often perform better on domains of different complexities and tendencies. Therefore, as well as having the facility to determine the best planning tool for each problem, it is also important to study the development of AP tools that can be applied to a wide range of different problems. This will significantly improve their ability to solve complex, real-world problems.

A main aim of this paper is to increase interest in applying automated planning to metrological processes. It is the authors' intention to apply the proposed technology to a broad range of applications, through which both the theoretical approach of using automated planning as well as the produced measurement plans can be validated. However, for this to be possible suitable tools and guidelines need to be made available for metrologists to use. The future challenge will be in developing tools that are useful for as broad a range of metrology planning problems as possible without the requirement of specific AP knowledge.

Keywords: Automated Planning; Measurement Uncertainty; Optimisation