1- INTRODUCTION

1.1- General

We study a problem in object-oriented knowledge representation. The traditional class-instance and inheritance paradigms form a sound basis for a computer simulation of real-world objects "as they are described", thereby giving a tool for expressing knowledge about the world in a specific way which is neither procedural nor declarative, and which we propose to call "object-oriented". In our opinion, knowledge representation differs from programming in that it has to be practised by experts of the domain and not by professional programmers. Hence, the programming tools it uses must be designed to fit the intellectual processes of the domain expert rather than to suit the needs of the implementer. In this respect, we consider that the class-instance mechanism is a very satisfactory machine realization of the "general concept"/"specific instance" way of thinking, whereas inheritance (simple or multiple) is far less acceptable as a classification scheme. As a consequence, we shall concentrate on improving the instanciation process and use inheritance in a standard way purely as a programming tool.

Anyhow, this well-known, well-implemented and well-understood paradigm must be extended in at least two ways in order to become a really usable tool for representing substantial amounts of knowledge about complex objects. Namely, it has to deal with two dimensions of structural complexity. First, objects usually must be considered from various points of view. Second, many objects are thought of as being composed of various parts that are themselves considered as sub-objects (and not as attributes). These two dimensions have been repeatedly explored in the past (Points of view : Goldstein & Bobrow in PIE [9], Bobrow & Stefik in LOOPS [2] and lately Carré in ROME [5] [6]. Part-whole : the LOOPS and Thinglab [3] systems, lately Blake & Cook [1] ). But nowhere have they been treated together: for instance the LOOPS primitives dealing with points of view (classes Node and Perspective) are not easily combined with metaclass Template catering for part-whole hierarchies.
1.2- Aim of paper

In this paper, we propose to bring these two dimensions together in an analysis and an implementation based on a restricted application domain, that of robot representation. We claim that by restricting the field we are able to formulate some proposals about the intellectual processes used by domain experts and thus to motivate our solutions.

More precisely, the aim of this paper is to study both dimensions together and to propose a set of tools which insures their harmonious cooperation. Our proposal extends the approach of Goldstein & Bobrow in PIE [9] on the multi-facet aspect by integrating the part-whole hierarchy aspect.

Fig. 1: A view of manipulator Previ as represented in Systalk