A task timeline is composed of multiple tasks or "activities", each with its interface channel requirements, which may occur once, repeatedly or continuously throughout the critical mission segment. Subject Matter Experts (SMEs) provided data for defining activities and assembling them into task timelines. For the attack/scout helicopter analysis, we first defined four critical mission segments, each containing multiple tasks for the two crewmembers and cockpit automation, as well as world events. The goal of these timelines was not strict accuracy in modeling events during mission performance, but rather to create a plausible testbench to evaluate different candidate cockpit designs. This motivation leads to many compromises in model development, as discussed below.

For many task steps, it is impossible to say when, precisely, the step will take place. This is especially true of "continuous" tasks such as those involved in flying the aircraft or monitoring aircraft subsystems (e.g., fuel status). Tasks of this nature must be done "continuously," but the physical resources used to, for example, fly the aircraft, may admit "disengagements" of up to several seconds in some circumstances (e.g., hands off stick, eyes removed from flight displays, etc.) Modeling tasks of this sort has traditionally been a problem for approaches to workload prediction, since the scheduling of these tasks is partially under operator control and permits various workload management strategies. By focusing on the problem of evaluating alternative cockpit configurations, we eliminate the need to be overly concerned with *when* these tasks are performed. Instead, we can assume an unrealistic or worst case frequency of task steps to serve as a "background" against which to evaluate conceptual crew stations. Although we know this produces an unrealistically high *absolute* estimate of conflict in the results of our simulations, as long as we use the same pattern of task steps in evaluating alternative crew stations, those designs which yield lower *relative* conflict values will generally produce better human-machine performance than those which yield higher conflict levels.

W/Index requires a static, single-path timeline (consisting only of start and stop times for all tasks or activities) for a single operator. The timeline may (in fact, it is expected to)

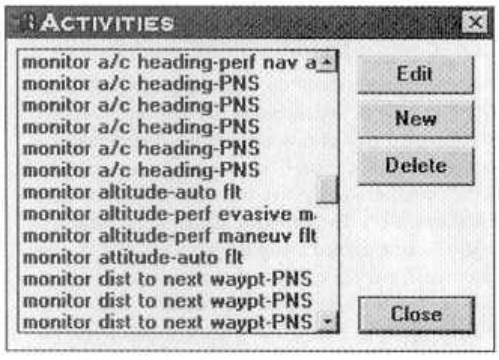

Figure 2a. W/Index list of previously defined activities for the helicopter scenarios.

represent the performance of multiple tasks in parallel, but unlike the partially-ordered graphs and sequential dependencies represented in MicroSAINT, or the alternative workload management strategies in CREWCUT which permit multiple paths through a task "network", W/Index permits no branching logic. Of course, multiple paths through a task network can each be modeled and run as separate task timelines with comparatively little effort in W/Index. The W/Index listing activities defined for the helicopter study is presented in Figure 2a while the screen for defining a new activity (reached by selecting "New" from the Activities screen in Fig 2a) is shown in Figure 2b. Note that the Edit Activity screen allows the definition of the activity in terms of the cockpit

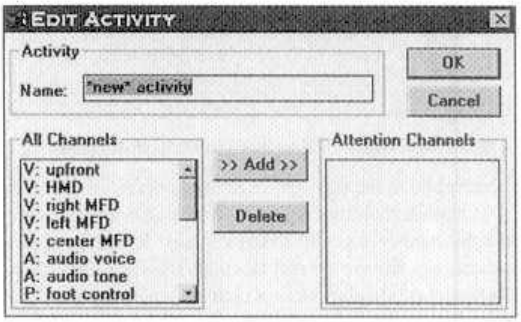

Figure 2b. W/Index Activity Definition screen.

channels which will be used whenever that activity is ongoing. The creation of channels and linking them to activities will be discussed in the next section below.

Once all needed activities have been defined, a timeline is created by assigning start and stop times for each instance of each activity which will occur during the timeline. Figure 3 shows the timeline creation and editing window in W/Index. Previously defined activities can be selected by pulling down the scrolling window in the "Edit Instance" frame, and then a start and stop time must be assigned to that instance of the activity. Figure 3 shows that the activity "monitor a/c heading-perf nav" has been selected and assigned a start time of .750 seconds into the scenario and a stop time of 1.750. Note that the timeline being constructed is presented in a scrolling frame at the bottom of the Time Line window. Instances of a previously defined activity can be added or deleted from the existing timeline and, as the timeline is built or modified, it can be saved via this window.

3.2.3 Interface/Activity Matrix

Each activity must also be assigned resource channels which the human operator will be required to use whenever that task is active. Resource channels