Hierarchical Grouping in Artificial Intelligence

Mark Mortimer

Lockheed Martin Canada Inc.
3001 Solandt Road
Kanata, Ontario K2K 2M8
mark.mortimer@lmco.com

Abstract

In designing a complex project that uses artificial intelligence often there is an inability to mesh immediate, local needs with larger, global needs. A technique to resolve this is to divide the project into hierarchical groups. With this, the lowest level groups within the hierarchy undertake most of the interactions with the environment, i.e. the immediate local needs. A higher level group looks at the overall situation of the environment and based on this, provides general directives or undertones to the lower levels. Using this design technique, the lower level groups can operate fairly continuously, in an independent manner while still being in harmony with the goals and objectives of the higher levels. The following expands on this technique and demonstrates its use through two case studies.

Introduction

Most problems which use artificial intelligence have a level of difficulty that has tempered the employment of it. Though some results, like automatic chess players, appear successful, these on the whole take advantage of a database of predetermined, pre-weighted actions. There is little reasoning as there are no unknowns in the chess environment (i.e. a required 8x8 board with a set number of select pieces). Reasoning becomes necessary when the problem includes an environment that is variable and ever changing. Reasoning is also necessary when the problem includes actions that are not preselected nor pre-weighted. The amount of reasoning and the need to make all the reasoning harmonious is why the employment of artificial intelligence is tempered.

Hierarchical grouping is a design technique that decreases the level of difficulty for such a problem, hence making artificial intelligence more employable. The hierarchical groups each contain a limited number of characteristics or actions. The group bases its limited set of actions on the current situation of the environment and its overall goals so that it can successfully deal with its sensory input. This allows for broader reasoning abilities and less use of databases, however there is still the need for a domain expert to dictate the reasoning of the groups.

Case Studies

The following are two case studies. They are written to highlight the hierarchical grouping design technique. The two case studies were chosen to demonstrate disparate cases that use the same technique. There is no warranty as to the authenticity or accuracy of the case studies.

Automatic Response Systems

For an automatic response system to make reasonable decisions (according to the domain expert), it must successfully deal with all situations, even those not predicted in advance. Though this appears daunting, the problem becomes manageable by reducing it to a series of hierarchical groups.

The case study involving an automatic response system is for a system of a naval warship. The ship has sensors (radars) that provide somewhat limited and unpredictable information regarding the situation surrounding the warship. Hence the situation is not predetermined. The ship also has weapon systems that defend it against threats. Last, to bring this together, the ship has an objective or overall goal that directs the ship's actions. The artificial intelligence in this problem takes the information from the sensors, evaluates it according to the ship's objective and when necessary, automatically respond with the ship's weaponry. This case study is manageable with hierarchical grouping.

The domain expert decided on four hierarchical groups for the automatic response system for the warship. The lowest level in the group is the device manager. Its limited actions involve communicating with all the devices (sensors and weapons). The purpose of the communication is dictated by its superior group, the tactical manager. There are only limited actions as the devices have only limited possible actions (e.g. a weapon to fire).

The device manager's superior group, the tactical manager, has a limited set of actions that consist of orchestrating a response against a threat. It uses the information received from the sensor input (e.g. range, bearing) to optimize the weaponry usage. Optimization includes setting the start time, possible repetition time and the end time or condition. The tactical manager uses its predefined knowledge of the weaponry also to estimate the threat's response. If the threat's measured response is not as predicted, then the tactical manager will cease use of the weapon and notify its superior of the failure. Its superior is the engagement manager.

The goal of the engagement manager's limited set of actions is to defeat a threat. The engagement manager uses the sensor's information regarding a threat and, based somewhat on a predetermined database, will allocate necessary weaponry to defeat the threat. The engagement manager can set up a series of possibly multiple weaponry actions that provide the greatest overall likelihood of defeating the threat. Also, if the tactical manager informs the engagement manager that a given tactic is not working, then the engagement manager can create a new series of multiple weaponry actions that do not include the failed weapon. The engagement manager will constantly have knowledge of the progress it has made in defeating the threat. This state of this progress is used by the engagement manager's superior, the battle manager.

The goal of the battle manager's limited set of actions is to assess the combat scenario and coordinate simultaneous engagements in order to achieve the ship's overall goal or objective. As there are multiple types and instances of weaponry onboard a warship, the battle manager must prevent overkill or underkill. It must also coordinate the weaponry in the engagement so that the use of one weapon does not adversely affect the use of another. In addition, the battle manager examines the combat scenario to ensure that its actions do not unduly affect other friendly forces. For example, a ship would not want to shoot a chaff round close to another friendly naval vessel (as a decoyed threat would then hit the other friendly vessel). The undertone for all the battle manager's decision however is its need to achieve the ship's overall goal.

The ship's overall goal is variable and can change with time. It may have a goal of self-survival, coastal bombardment, and / or convoy escort. However, the battle manager uses this goal to provide any directives to its immediate inferior, the engagement manager. For example, the battle manager may have reasoning that indicates that self-destruction is better than losing a (higher valued) convoy member hence, this undertone, a result of the ship's overall goal, would be part of the battle manager.

By using these four levels of hierarchal grouping, we have successfully divided a complex automatic response system problem into manageable groups. Further, due to the communication between the groups, the high level group that assesses the ship's overall goal provides undertones to inferior groups so that low level actions are influenced by the higher levels. For the system as a whole, the weapon usage is optimized and includes the strategic directives of the ship.

Guiding a Collection of Units

The second case study using hierarchical grouping is for guiding a collection of units. The challenge in guiding such a collection is the need to trade off the goals of the individual unit against the goals of the collection. The particular case study for demonstrating this technique for guiding units is for a military unit within a collection of military units, i.e. a military simulation. Remember that this case study is meant to illustrate a coding technique and the subject is not necessarily authentic or accurate.

In a traditional military simulation or wargame there are collections of units that act based on; the current situation, how the current situation arose, how its actions may affect the situation and how other's actions may affect the situation. In a typical wargame, there are many units of diverse types that move within a complex, variable environment. A unit solely choosing an action from a database suffers from an inability to adapt to unique situations. However, using hierarchical groups, the units in a wargame can have artificial intelligence that supplies valid directives for all environments, hence making all actions reasonable.

For a unit's actions to be reasonable, the decision that it takes must have an objective or goal. In the previous case study, the ship's objective or goal was the undertone for the decision making. In much the same manner, a military unit must have an undertone. This undertone should not prevent the unit from operating independently, however, it should influence the units decision. The following illustrates hierarchical groupings that carry the undertone to all groups.

The lowest hierarchical level in the military wargame is responsible for doing the actions (much as the device manager did in the previous case study). This lowest level could simulate a soldier, a platoon, or a company. This lowest level has a limited number of actions it can perform. Though it has some of its own reasoning, it also receives direction from its superior group. If the lowest level were a soldier, then the soldier would consider its immediate situation, prior actions and postulated activities. Considerations include the presence of enemy soldiers, daylight hours, and / or the amount of remaining ammunition. It then acts based on the immediate situation while also considering the undertone provided by its superior level

The superior level is a construct used by the artificial intelligence. It is not an entity and can not perform any physical actions that affect the environment. Its purpose is to provide directives that affect its own immediately subordinate level. For example, consider a simulated platoon. The platoon is a collection of soldiers. The platoon does not fight as it does not have any physical capability. Rather, it guides its collection of soldiers to achieve a goal that (presumably) one soldier would not be able to achieve on its own. The platoon would assess the situation in its sphere of influence and inform or direct its soldier to achieve a certain goal or objective; for instance, attack in a general given direction or capture a house.

It is then the responsibility of the soldier to take this information or directive and use it to guide its actions. The soldier may not instantly follow the directive (eg first resolving an ongoing fire-fight) or the soldier may have a preferential task immediately at hand (eg disarming a group of prisoners). However, the soldier would have a set of decision making parameters (aka threshold level units) that would dictate when it is to follow its own directives or when it is to follow the information passed down from its superior or when it follows some combination of each.

This design technique using hierarchical grouping does not have a limit in the depth of groups. The above example is for soldiers and a platoon. There could be many platoons that are grouped into a company, the companies are grouped into battalions, then brigades, and so on. The construct resembles a typical tree structure with a root node or group (the highest level) and connecting branches to lower levels or inferior nodes or groups.

Further, there is no limit to the breadth of the groups (aside from computer hardware limits). There can be as many soldiers, platoons and companies as necessary. The number and size of the groupings are typically resolved in scenario test phases.

Thus with the design technique of hierarchical grouping, there is an efficient technique that resolves the complex problem of directing a unit within a collection of units. By dividing the decision making into hierarchical groups or levels, the war game simulation becomes manageable. The lowest group of the simulation is the group that does actions and affects the world about it. The higher levels are able to obtain more information about the situation for a wider field. They then provide the information as directives or undertones that the lower levels consider but not necessary immediately completely adhere to.

Constraining Communications

A beneficial consequence of using hierarchical grouping is that there is limited communication between the groups. The only communication is between an inferior and its own superior. Further, this communication is constrained to be the superior influencing the objective or goal for the inferior level by inserting undertones. The inferior level is constrained by providing the status of its progress in achieving an objective or goal. This is typical of a military ‘chain of command’ limited communication.

Ease in Programming

The above description of the hierarchical grouping design technique is purposefully written to display the design technique’s similarity to designing object oriented programs. By interchanging the word ‘group’ in much of the preceding with the word ‘class’, many features of object oriented programming are evident. For example on replacement we see classes and sub-classes with constrained communication. Each class has a limited set of actions. Each class has its own view of the situation and its own functions on what to do as based on its view. This is similar to discussions in many texts on object oriented programming.

For the military simulation case study, both C and C++ computer language constructs are used. This is due to the evolution of the code rather than any foresight. The groups are coded as classes. There is constrained communication between the classes. However, rather than soldiers being the lowest level, the lowest level is a corp. The next superior level(or group) is the army and the final superior or root level is the theatre group.

For the automated response system, the groups are coded as knowledge sources in an expert system environment. The knowledge sources are data driven, where effectively, all the data of the situation is situated on a blackboard. Because of many benefits of using an expert system with a blackboard, standard object oriented programming was not used.

Other Applications

The design technique of hierarchical grouping is applicable to many artificial intelligence areas. The case study of having a unit in a collection of units is purposefully vague. The unit can be a character in a role playing game, it can be a bee in a colony, or even a staff member of a typical business.

Another application of hierarchical grouping is in varying the amount of independence of sub-groups. For example, varying the independence in a military simulation makes the simulation representative of various societies. Sub-groups that allow a high degree of independence can be construed to resemble typical western forces in the historical red star/white star scenarios. Eastern forces would have sub-groups with much less independence and would thus have most of the decision making occurring at the root node. By running simulations using various degrees of independence in sub-groups, resulting effects can be observed.

At present, the analysis with the military simulation case study is looking at fixing the necessary and sufficient limited actions for each group. Once reasonable actions are achieved, then the effects of varying the degree of independence will be studied.

There can also be some communication for sub-groups within the same group. For instance, soldiers can communicate with soldiers of their own platoon, their platoon communicates with its fellow platoon members of their company, and so on. This intra-group communication is possible throughout the depth of the hierarchy.

Conclusion

With hierarchical grouping, a complex project can have reasonable artificial intelligence. Hierarchical grouping has a low level that does the interactions with the environment. Possible interactions consist of initiating automatic response systems or moving individual units within a collection. Superior levels add undertones to lower levels' interactions. The undertones are based on the superior level's larger view of the situation. The undertone will influence the lower level without dictating the lower level decision making. There is no limit to the number of levels or depth of grouping nor in the breadth or number of elements within a group. Thus, through hierarchical grouping, there is a design technique for complex projects that provides reasonable artificial intelligence.