DeepX

Tommy Petersen
tp@aia.dk

December 23, 2015

Abstract

This article presents a theory for an agent based on combining simple extensions of explorative agents.

1 Introduction

In "http://www.ai-agents.com/PDF/ShallowX/shallowX.pdf", an agent is described called **ShallowX** which is an explorative agent. This article presents a theory for combining simple extensions of ShallowX agents into a larger agent. As with ShallowX, the theory can also be used as a design framework for a general agent. Both the theory and the general agent are called **DeepX**.

2 ShallowX recap

A diagram representing the general data flow around ShallowX is shown in figure 1.

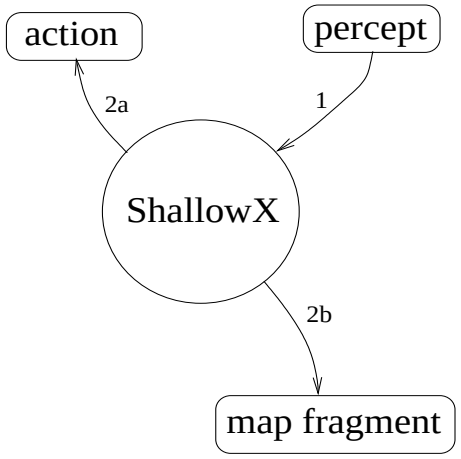

Figure 1: ShallowX data flow diagram.

The data named "**action**", "**percept**" and "**map fragment**" are all finite symbol sequences and should be understood as follows:

· percept Sensory data from the environment.

· action Chosen action.

· map fragment Part of a map of the environment. The map fragment is on the form (C, I), where C is a context, and I is information.

ShallowX continually acts and maps as a result of a never ending stream of percepts from the environment.

3 Initial combination

Using the diagram in figure 1 directly, an initial combination of the ShallowX agents is given in figure 2.

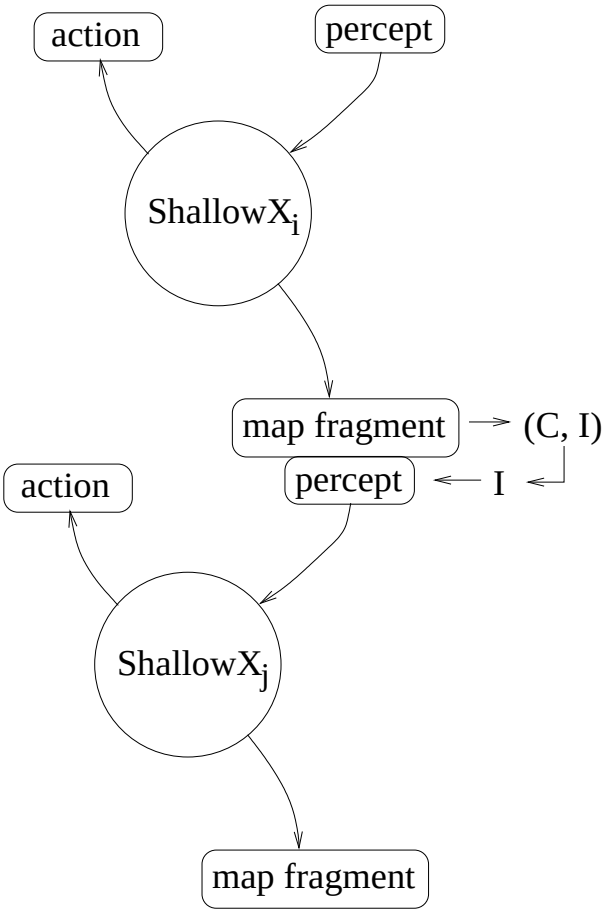

Figure 2: Initial ShallowX combination.

In figure 2, ShallowX$_j$ has been added as a client of ShallowX$_i$ receiving the latter's map fragment information, I. The client ShallowX$_j$ is added to a location adressed by the map fragment context, C.

ShallowX$_j$ is directly affected by ShallowX$_i$ via percepts (the I part of ShallowX$_i$'s map fragment), but cannot directly affect ShallowX$_i$ by it's actions.

4 Extended combination

ShallowX$_i$ might send a periodic signal to ShallowX$_j$ and if the latter detects this, then it will emit an action in an attempt to explore something else. But the action emitted from ShallowX$_j$ in figure 2 does not affect ShallowX$_i$ directly. In order to achieve this, ShallowX is extended to **ShallowX+** as illustrated in figure 3.

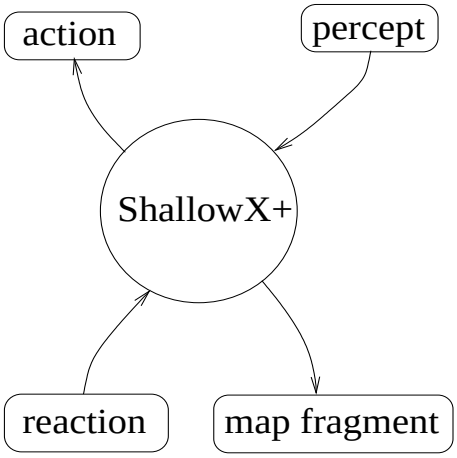

Figure 3: The extension of ShallowX to ShallowX+.

The added data named "**reaction**" is a finite symbol sequence and should be understood as being the client's reaction to the delivered map fragment.

Now the diagram in figure 3 can be used to combine the ShallowX+ agents as illustrated in figure 4.

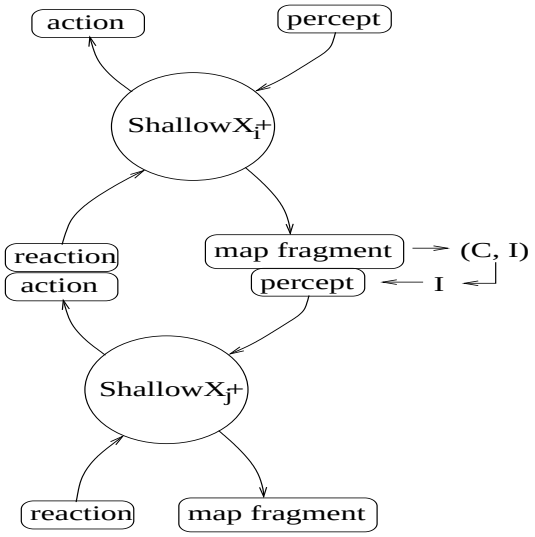

Figure 4: ShallowX+ combination.

4.1 Reaction location

If ShallowX$_j$+ has detected a possible periodic signal coming from ShallowX$_i$+ then ShallowX$_j$+ can now affect ShallowX$_i$+ by sending an action. To ShallowX$_i$+ this

action is seen as a reaction coming from ShallowX$_j$+. This reaction should affect ShallowX$_i$+ for as long as possible in order to stop the possible periodic signal sent to ShallowX$_j$+. Hence the reaction is located at the beginning of ShallowX$_i$+'s experience:

$$reaction.experience$$

The concatenation then becomes the new experience as in the following example:

$$reaction = 10$$

$$experience = 1101010001$$

$$reaction.experience = 10.1101010001$$

$$new\ experience = 101101010001$$

4.2 Asynchronous calls

After ShallowX$_i$+ has called ShallowX$_j$+ with a percept, it does not wait for ShallowX$_j$+ to return an action. The reason is that ShallowX$_j$+ might not return an action since there might not be an overlap in ShallowX$_j$+'s concatenated potential. Still, a "callback" link from ShallowX$_j$+ to ShallowX$_i$+ is created so that ShallowX$_j$+ can return it's action when it is ready—possibly after more than one percept. After the action is returned, the callback link is removed.

The calls are then asynchronous avoiding ShallowX+ agents blocking when they have sent map fragments to their clients.

4.3 Message loss

As figure 3 illustrates, a ShallowX+ agent can receive a percept from it's environment or it can receive a reaction from a client. After receiving input, ShallowX+ processes it and is not available for other inputs until the processing is completed returning a map fragment or an action. If other percepts or reactions are sent to ShallowX+ while it is processing, then these inputs are lost.

The main argument for accepting message loss is simplicity avoiding the need for extension by a queueing system.

It can also be argued, that if the environment is almost static then the input from the environment will be close to periodic. It is then possible that ShallowX+ can still detect such a period even though some messages are lost and hence choose an action in order to explore new parts of the environment. As for the map fragment, it may not be considered as correct, but it is still a map of what ShallowX+ perceives of the environment.

Ultimately, if a ShallowX+ agent is directly exposed to the physical environment then there will always be a loss of input and this observation can be used to argue in favor of allowing message loss across all ShallowX+ agents in DeepX.

4.4 Multiple callers

Consider figure 4 and observe that ShallowX$_j$+ can actually receive many percepts from ShallowX$_i$+ before it returns an action. The reason is that the first percepts received by ShallowX$_j$+ may not make it detect an overlap—that may require many percepts. But more than that, ShallowX$_j$+ can also receive percepts from other callers, ShallowX$_k$+, as illustrated in Figure 5.

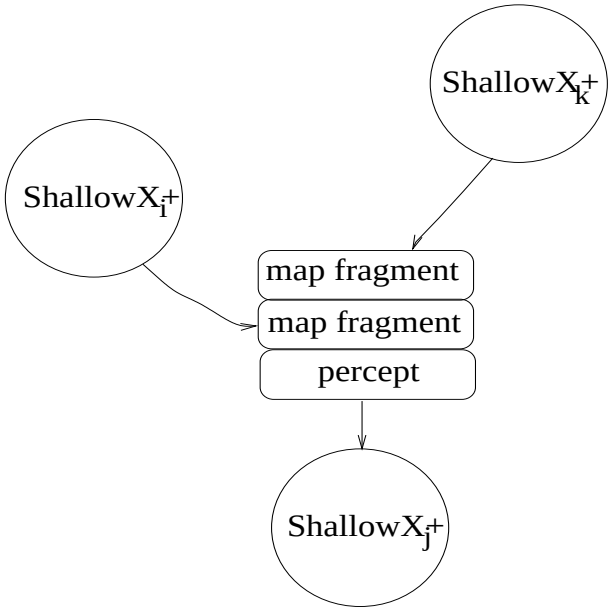

Figure 5: ShallowX+ with multiple callers.

The percept from ShallowX$_i$+ is received before the percept from ShallowX$_k$+. Callback links to both ShallowX$_i$+ and ShallowX$_k$+ are created in order to return ShallowX$_j$+'s action to both callers when the action is ready. After the action is returned, the links are removed. This sheme generalizes to arbitrarily many callers of ShallowX$_j$+.

The reason for using this scheme of multiple callers is to allow for different parts of DeepX to combine their map fragments into a single part that gets a more nuanced view of the environment. The single part can then also detect possible overlaps between the different parts that affect it and hence communicate this mutual overlap back.

5 Summary

The **DeepX** agent has been defined based on combinations of extensions of **ShallowX** agents. DeepX can be an arbitrarily deep network of explorative ShallowX+ agents.

6 Ideas for future work

The question is how the DeepX network of explorative agents will react to phenomenons of varying complexity occuring in the physical world. An implementation of DeepX is desired in order to help with an answer.