Logic engineering

The basic framework is quite general and can be refined in various ways to give us the properties appropriate for the intended applications. Logic engineering is the subject of engineering logics to fit new applications. It is potentially a very broad subject, drawing on all branches of logic, computer science and mathematics. In this chapter, however, we are restricting ourselves to the particular engineering of modal logics. We will consider how to re-engineer basic modal logic to fit the following readings of φ:

- It is necessarily true that φ
- It will always be true that φ
- It ought to be that φ
- Agent Q believes that φ
- Agent Q knows that φ
- After any execution of program P, φ holds.

As modal logic automatically gives us the connective $\Diamond$, which is equivalent to $\neg\Box\neg$, we can find out what the corresponding readings of $\Diamond$ in our system will be. For example, 'it is *not* necessarily true that *not* ϕ' means that it is possibly true that ϕ. You could work this out in steps:

It is *not* necessarily true that ϕ
= it is possible that *not* ϕ.

Therefore,

It is *not* necessarily true that *not* ϕ
= it is possible that *not not* ϕ
= it is possible that ϕ.

Let us work this out with the reading 'agent Q knows ϕ' for $\Box\phi$. Then, $\Diamond\phi$ is read as

agent Q does *not* know *not* ϕ
= as far as Q's knowledge is concerned, ϕ could be the case
= ϕ is consistent with what agent Q knows
= for all agent Q knows, ϕ.

$\Box\phi$	$\Diamond\phi$
It is necessarily true that ϕ	It is possibly true that ϕ
It will always be true that ϕ	Sometime in the future ϕ
It ought to be that ϕ	It is permitted to be that ϕ
Agent Q believes that ϕ	ϕ is consistent with Q's beliefs
Agent Q knows that ϕ	For all Q knows, ϕ
After any execution of program P, ϕ holds	After some execution of P, ϕ holds