Principles			Actionable Behavior	Example
Reification	*of a*	**command**	transforms a command into a **persistent, interactive instrument**	**Instrument**
	of an	**effect**	transforms effects into **a persistent, interactive substrate** that **contains objects, interprets objects** and **manages relationships among objects.**	**Substrate**
Polymorphism	*of a*	**command**	enables an instrument to affect **different types of objects**	**Multi-object**
	of an	**effect**	enables a substrate to manage **different types of relationships**	**Multi-relationship**
Reuse	*of a*	**command**	applies **previous actions to different objects**	**Macro**
	of an	**effect**	applies **previous effects to different objects**	**Template**
Discovery	*of a*	**command**	lets users see **possible future actions**	**Feedforward**
	of an	**effect**	lets users see how the system **interpreted their behavior**	**Feedback**
Expressivity	*of a*	**command**	lets users transform **input deviation from the norm into rich output**	**Input variation**
	of an	**effect**	lets users **fine-tune effects into reusable objects**	**Tweak**
Customizability	*of a*	**command**	lets users create instruments or **transform properties into families of tools**	**Currying**
	of an	**effect**	lets users redefine the mapping between actions and effects	**Remapping**
Appropriability	*of a*	**command**	lets users **reuse properties for different purposes**	**Technical reasoning**
	of an	**effect**	lets users **reinterpret relationships among objects and substrates**	**Redefinition**