KomaMRIBase

```
struct Phantom{T}
  # x0, y0, z0, T1, T2, PD...
  motion::Union{NoMotion, Motion{T}, MotionList{T}}
end
```

```
struct MotionList{T} <: AbstractMotion{T}
  motions::Vector{<:Motion{T}}
end
```

```
struct Motion{T}
  action::AbstractAction{T}
  time::AbstractTimeSpan{T}
  spins::AbstractSpinSpan
end
```

Translate
Rotate
HeartBeat
Path
FlowPath

TimeRange
Periodic

$t_{unit}(t)$
1
...
t_{start}
t_{end}
t
$t_{unit}(t)$
1
...
period
t

AllSpins
SpinRange

```
function get_spin_coords(
 ml::NoMotion, x,y,z,t
) where {T<:Real}
       return x,y,z
end
```

```
function get_spin_coords(
 ml::Motion{T}, x,y,z,t
) where {T<:Real}
       # ...
end
```

```
function get_spin_coords(
 ml::MotionList{T},x,y,z,t
) where {T<:Real}
       # ...
end
```

KomaMRICore

```
function run_spin_excitation!(
    p::Phantom{T},
    seq::DiscreteSequence{T},
    sig::AbstractArray{Complex{T}},
    M::Mag{T},
    ...
) where {T<:Real}
   # ...
   get_spin_coords(p.motion, p.x, p.y, p.z, seq.t)
   # ...
end
```

```
function run_spin_precession!(
    p::Phantom{T},
    seq::DiscreteSequence{T},
    sig::AbstractArray{Complex{T}},
    M::Mag{T},
    ...
) where {T<:Real}
   # ...
   get_spin_coords(p.motion, p.x, p.y, p.z, seq.t)
   # ...
end
```