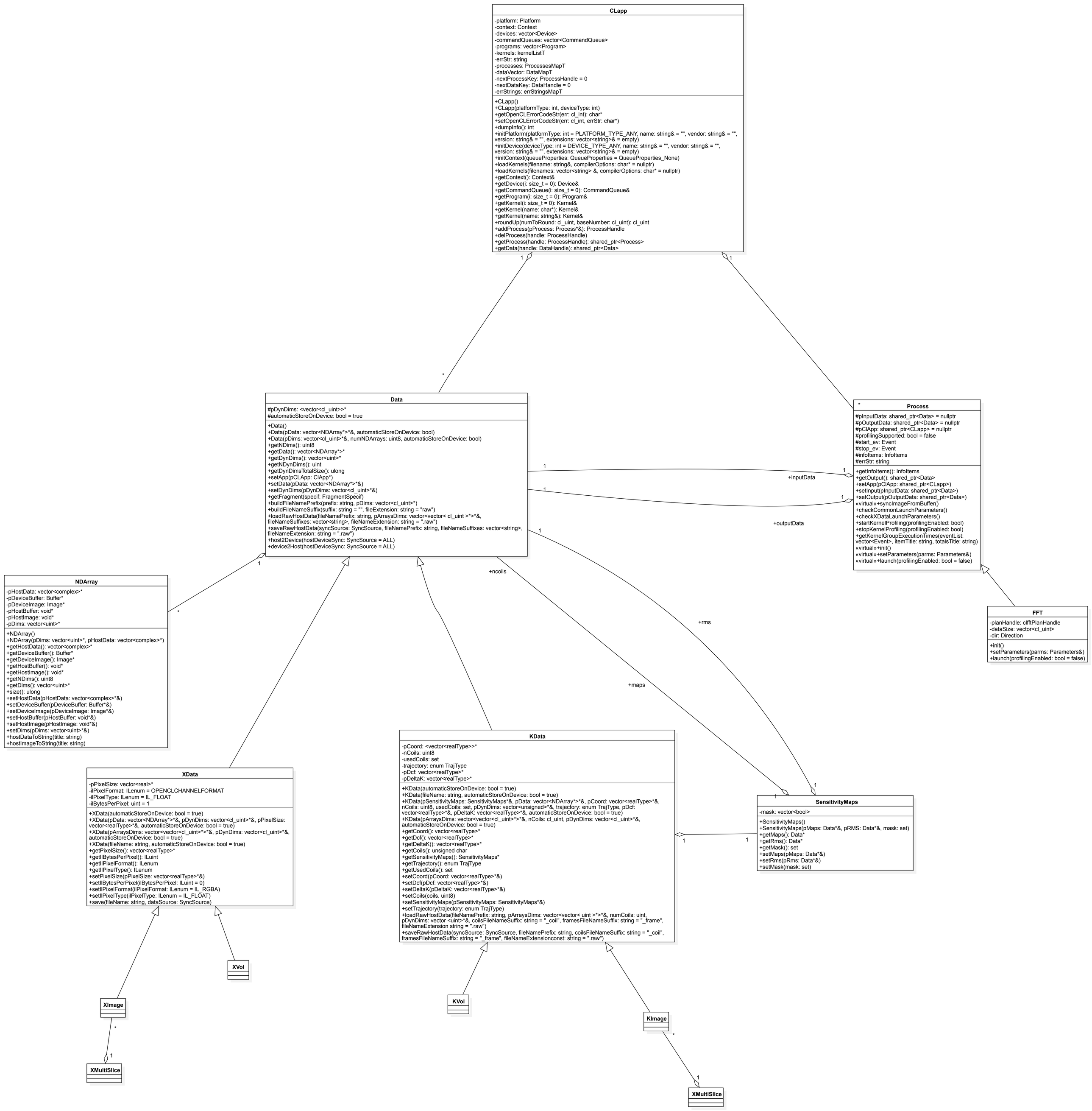
CLapp
-platform: Platform
-context: Context
-devices: vector<Device>
-commandQueues: vector<CommandQueue>
-programs: vector<Program>
-kernels: kernelListT
-errStr: string
-processes: ProcessesMapT
-dataVector: DataMapT
-nextProcessKey: ProcessHandle = 0
-nextDataKey: DataHandle = 0
-errStrings: errStringsMapT
+CLapp()
+CLapp(platformType: int, deviceType: int)
+getOpenCLErrorCodeStr(err: cl_int): char*
+setOpenCLErrorCodeStr(err: cl_int, errStr: char*)
+dumpInfo(): int
+initPlatform(platformType: int = PLATFORM_TYPE_ANY, name: string& = "", vendor: string& = "", version: string& = "", extensions: vector<string>& = empty)
+initDevice(deviceType: int = DEVICE_TYPE_ANY, name: string& = "", vendor: string& = "", version: string& = "", extensions: vector<string>& = empty)
+initContext(queueProperties: QueueProperties = QueueProperties_None)
+loadKernels(filename: string&, compilerOptions: char* = nullptr)
+loadKernels(filenames: vector<string> &, compilerOptions: char* = nullptr)
+getContext(): Context&
+getDevice(i: size_t = 0): Device&
+getCommandQueue(i: size_t = 0): CommandQueue&
+getProgram(i: size_t = 0): Program&
+getKernel(i: size_t = 0): Kernel&
+getKernel(name: char*): Kernel&
+getKernel(name: string&): Kernel&
+roundUp(numToRound: cl_uint, baseNumber: cl_uint): cl_uint
+addProcess(pProcess: Process*&): ProcessHandle
+delProcess(handle: ProcessHandle)
+getProcess(handle: ProcessHandle): shared_ptr<Process>
+getData(handle: DataHandle): shared_ptr<Data>
1
1
*
Data
#pDynDims: <vector<cl_uint>>*
#automaticStoreOnDevice: bool = true
+Data()
+Data(pData: vector<NDArray*>*&, automaticStoreOnDevice: bool)
+Data(pDims: vector<cl_uint>*&, numNDArrays: uint8, automaticStoreOnDevice: bool)
+getNDims(): uint8
+getData(): vector<NDArray*>*
+getDynDims(): vector<uint>*
+getNDynDims(): uint
+getDynDimsTotalSize(): ulong
+setApp(pCLApp: ClApp*)
+setData(pData: vector<NDArray*>*&)
+setDynDims(pDynDims: vector<cl_uint>*&)
+getFragment(specif: FragmentSpecif)
+buildFileNamePrefix(prefix: string, pDims: vector<cl_uint>*)
+buildFileNameSuffix(suffix: string = "", fileExtension: string = "raw")
+loadRawHostData(fileNamePrefix: string, pArraysDims: vector<vector< cl_uint >*>*&, fileNameSuffixes: vector<string>, fileNameExtension: string = ".raw")
+saveRawHostData(syncSource: SyncSource, fileNamePrefix: string, fileNameSuffixes: vector<string>, fileNameExtension: string = ".raw")
+host2Device(hostDeviceSync: SyncSource = ALL)
+device2Host(hostDeviceSync: SyncSource = ALL)
1
1
+inputData
1
1
+outputData
*
Process
#pInputData: shared_ptr<Data> = nullptr
#pOutputData: shared_ptr<Data> = nullptr
#pClApp: shared_ptr<CLapp> = nullptr
#profilingSupported: bool = false
#start_ev: Event
#stop_ev: Event
#infoItems: InfoItems
#errStr: string
+getInfoItems(): InfoItems
+getOutput(): shared_ptr<Data>
+setApp(pClApp: shared_ptr<CLapp>)
+setInput(pInputData: shared_ptr<Data>)
+setOutput(pOutputData: shared_ptr<Data>)
«virtual»+syncImageFromBuffer()
+checkCommonLaunchParameters()
+checkXDataLaunchParameters()
+startKernelProfiling(profilingEnabled: bool)
+stopKernelProfiling(profilingEnabled: bool)
+getKernelGroupExecutionTimes(eventList: vector<Event>, itemTitle: string, totalsTitle: string)
«virtual»+init()
«virtual»+setParameters(parms: Parameters&)
«virtual»+launch(profilingEnabled: bool = false)
1
+ncoils
1
+rms
+maps
1
1
NDArray
-pHostData: vector<complex>*
-pDeviceBuffer: Buffer*
-pDeviceImage: Image*
-pHostBuffer: void*
-pHostImage: void*
-pDims: vector<uint>*
+NDArray()
+NDArray(pDims: vector<uint>*, pHostData: vector<complex>*)
+getHostData(): vector<complex>*
+getDeviceBuffer(): Buffer*
+getDeviceImage(): Image*
+getHostBuffer(): void*
+getHostImage(): void*
+getNDims(): uint8
+getDims(): vector<uint>*
+size(): ulong
+setHostData(pHostData: vector<complex>*&)
+setDeviceBuffer(pDeviceBuffer: Buffer*&)
+setDeviceImage(pDeviceImage: Image*&)
+setHostBuffer(pHostBuffer: void*&)
+setHostImage(pHostImage: void*&)
+setDims(pDims: vector<uint>*&)
+hostDataToString(title: string)
+hostImageToString(title: string)
*
FFT
-planHandle: clfftPlanHandle
-dataSize: vector<cl_uint>
-dir: Direction
+init()
+setParameters(parms: Parameters&)
+launch(profilingEnabled: bool = false)
XData
-pPixelSize: vector<real>*
-ilPixelFormat: ILenum = OPENCLCHANNELFORMAT
-ilPixelType: ILenum = IL_FLOAT
-ilBytesPerPixel: uint = 1
+XData(automaticStoreOnDevice: bool = true)
+XData(pData: vector<NDArray*>*&, pDynDims: vector<cl_uint>*&, pPixelSize: vector<realType>*&, automaticStoreOnDevice: bool = true)
+XData(pArraysDims: vector<vector<cl_uint>*>*&, pDynDims: vector<cl_uint>*&, automaticStoreOnDevice: bool = true)
+XData(fileName: string, automaticStoreOnDevice: bool = true)
+getPixelSize(): vector<realType>*
+getIlBytesPerPixel(): ILuint
+getIlPixelFormat(): ILenum
+getIlPixelType(): ILenum
+setPixelSize(pPixelSize: vector<realType>*&)
+setIlBytesPerPixel(ilBytesPerPixel: ILuint = 0)
+setIlPixelFormat(ilPixelFormat: ILenum = IL_RGBA)
+setIlPixelType(ilPixelType: ILenum = IL_FLOAT)
+save(fileName: string, dataSource: SyncSource)
KData
-pCoord: <vector<realType>>*
-nCoils: uint8
-usedCoils: set
-trajectory: enum TrajType
-pDcf: vector<realType>*
-pDeltaK: vector<realType>*
+KData(automaticStoreOnDevice: bool = true)
+KData(fileName: string, automaticStoreOnDevice: bool = true)
+KData(pSensitivityMaps: SensitivityMaps*&, pData: vector<NDArray*>*&, pCoord: vector<realType>*&, nCoils: uint8, usedCoils: set, pDynDims: vector<unsigned>*&, trajectory: enum TrajType, pDcf: vector<realType>*&, pDeltaK: vector<realType>*&, automaticStoreOnDevice: bool = true)
+KData(pArraysDims: vector<vector<cl_uint>*>*&, nCoils: cl_uint, pDynDims: vector<cl_uint>*&, automaticStoreOnDevice: bool = true)
+getCoord(): vector<realType>*
+getDcf(): vector<realType>*
+getDeltaK(): vector<realType>*
+getCoils(): unsigned char
+getSensitivityMaps(): SensitivityMaps*
+getTrajectory(): enum TrajType
+getUsedCoils(): set
+setCoord(pCoord: vector<realType>*&)
+setDcf(pDcf: vector<realType>*&)
+setDeltaK(pDeltaK: vector<realType>*&)
+setCoils(coils. uint8)
+setSensitivityMaps(pSensitivityMaps: SensitivityMaps*&)
+setTrajectory(trajectory: enum TrajType)
+loadRawHostData(fileNamePrefix: string, pArraysDims: vector<vector< uint >*>*&, numCoils: uint, pDynDims: vector <uint>*&, coilsFileNameSuffix: string = "_coil", framesFileNameSuffix: string = "_frame", fileNameExtension string = ".raw")
+saveRawHostData(syncSource: SyncSource, fileNamePrefix: string, coilsFileNameSuffix: string = "_coil", framesFileNameSuffix: string = "_frame", fileNameExtensionconst: string = ".raw")
1
1
SensitivityMaps
-mask: vector<bool>
+SensitivityMaps()
+SensitivityMaps(pMaps: Data*&, pRMS: Data*&, mask: set)
+getMaps(): Data*
+getRms(): Data*
+getMask(): set
+setMaps(pMaps: Data*&)
+setRms(pRms: Data*&)
+setMask(mask: set)
XVol
XImage
*
1
XMultiSlice
KVol
KImage
*
1
XMultiSlice