v2.1

GCX Engine Version 2.1

G(arbage) C(collected) X(Query) Engine

– Class Diagrams –

Michael Schmidt* Gunnar Jehl†

May 2009

Saarland University Database Group
Freiburg University Database Group

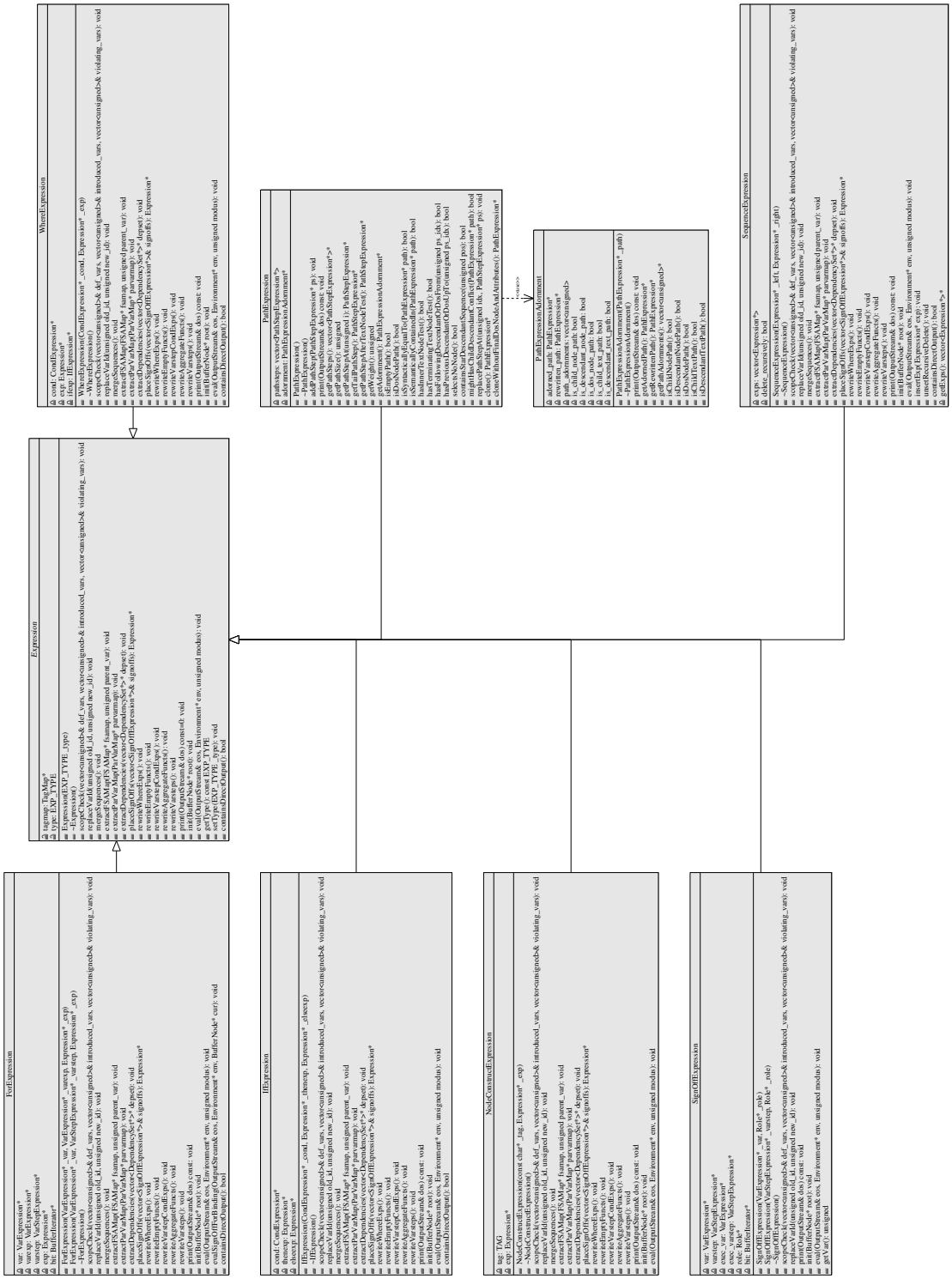
*mschmidt@informatik.uni-freiburg.de
†jehl@informatik.uni-freiburg.de

Abstract

The G(arbage) C(ollected) X(Query) engine is the first streaming XQuery engine that implements active garbage collection, a novel buffer management strategy in which both static and dynamic analysis are exploited. This technique actively purges main memory buffers at runtime based on the current status of query evaluation. This approach aims at both keeping main memory consumption low at runtime and speeding up query evaluation. For detailed information on active garbage collection in XQuery engines please visit the GCX project homepage at

http://dbis.informatik.uni-freiburg.de/index.php?project=GCX.

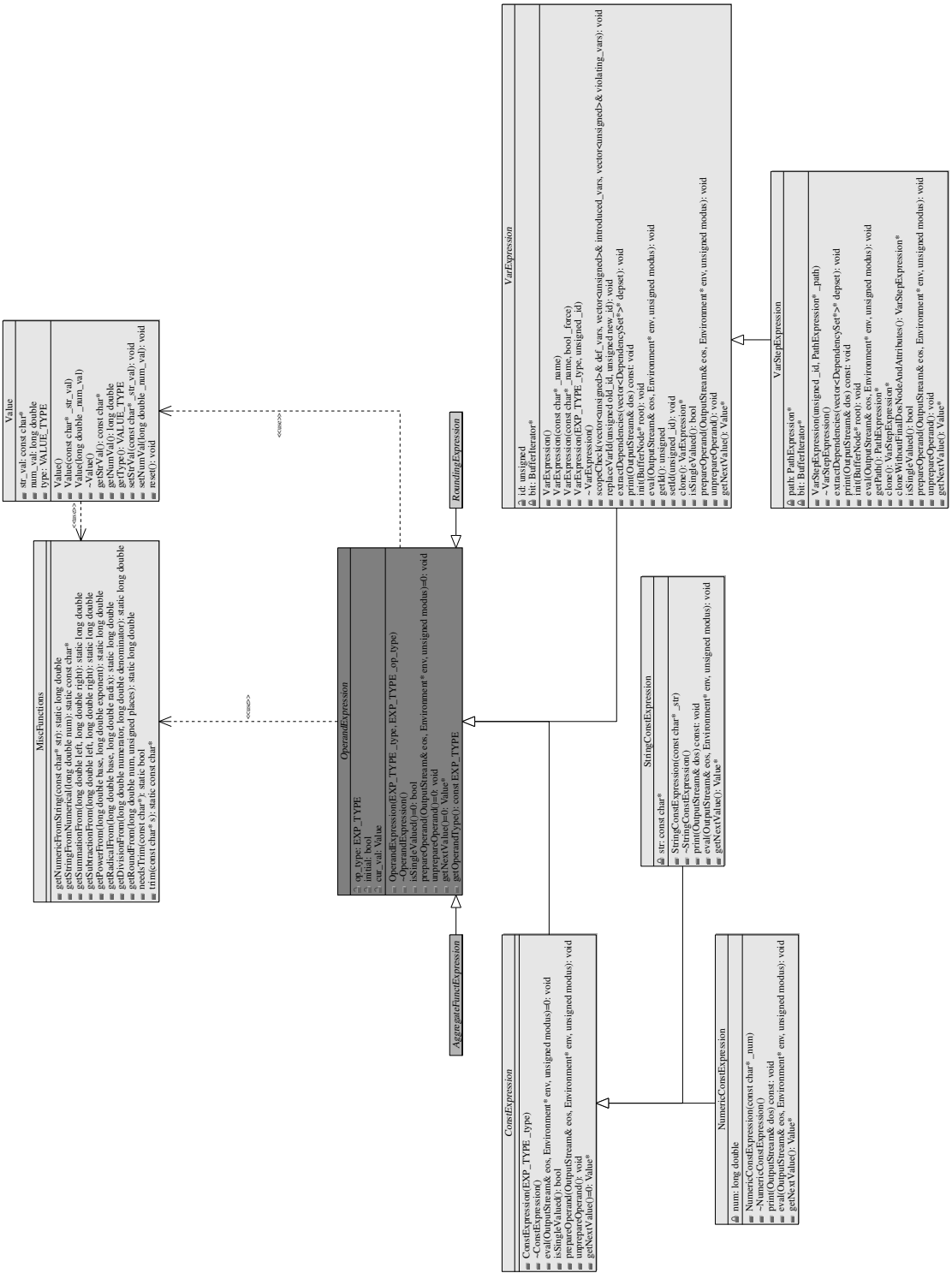
1 Class Diagram: Expressions



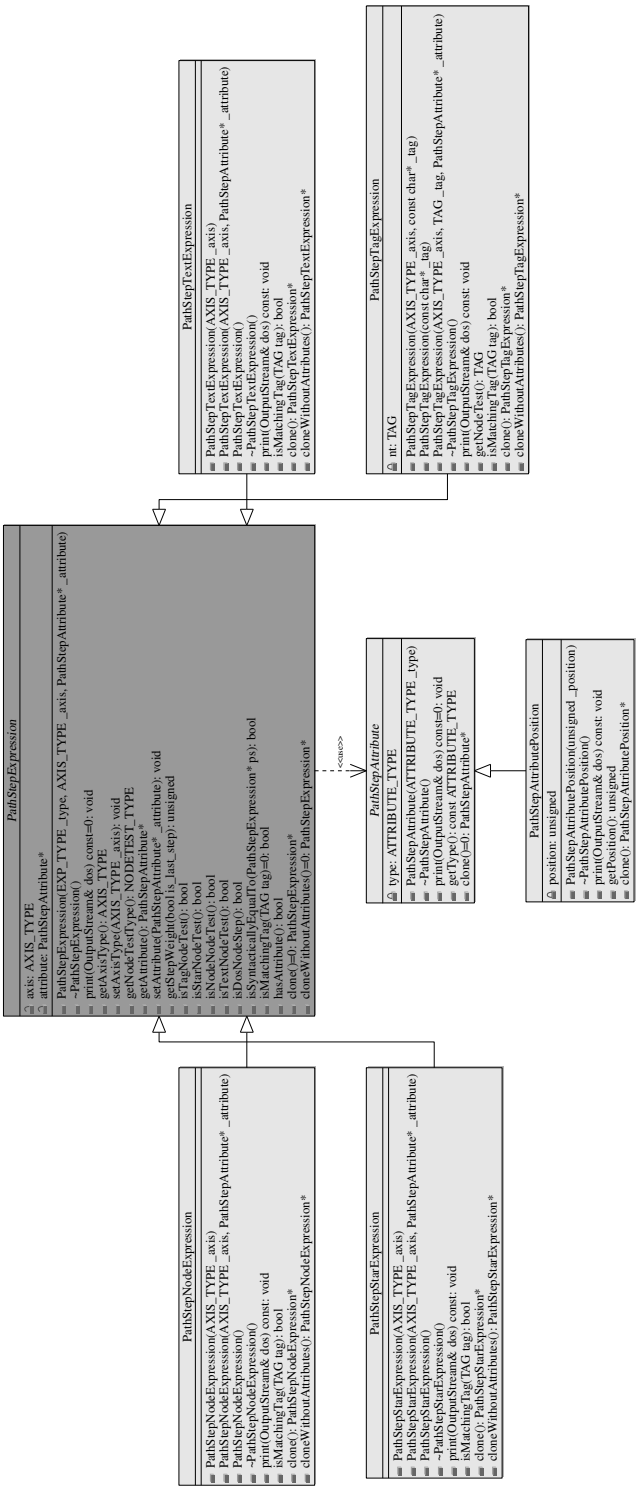
2 Class Diagram: Expressions (cont'd)

3 Class Diagram: Conditions

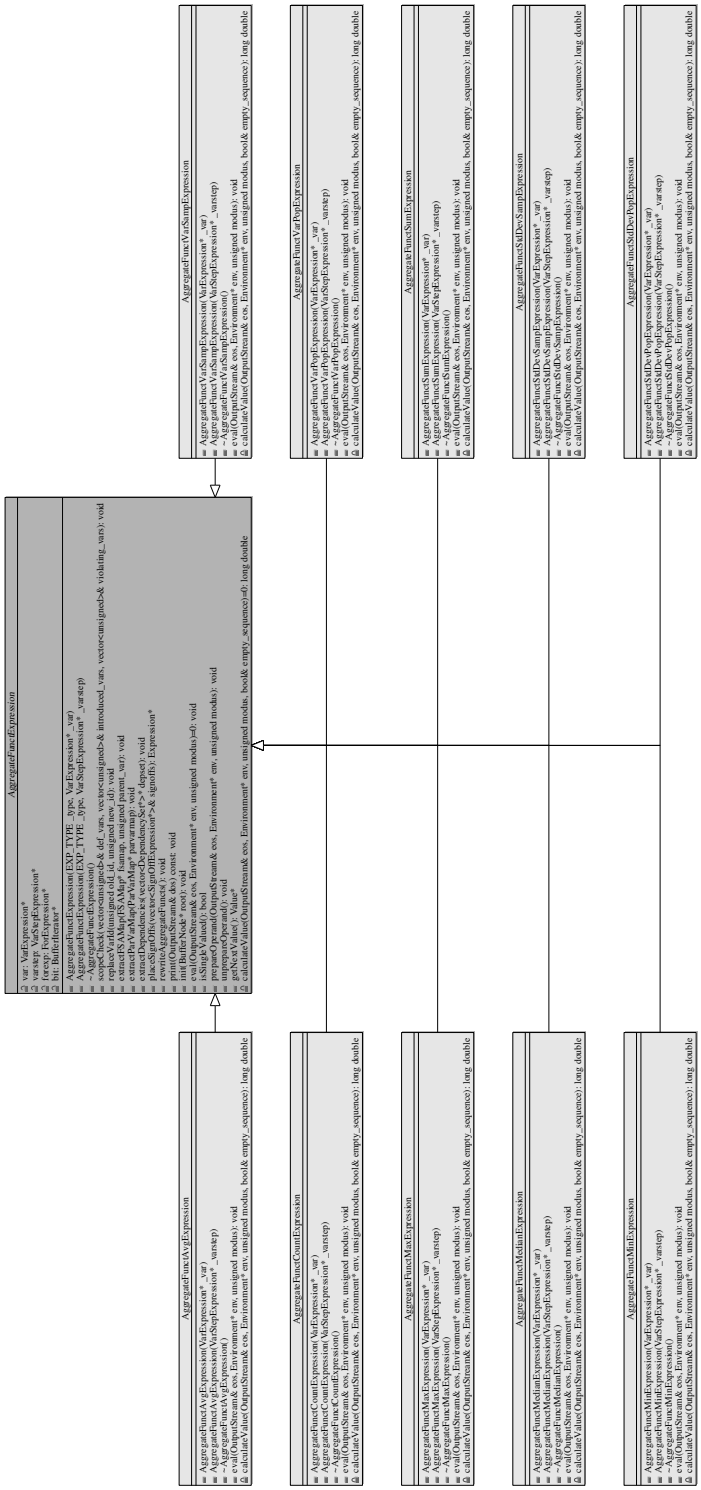
4 Class Diagram: Operands



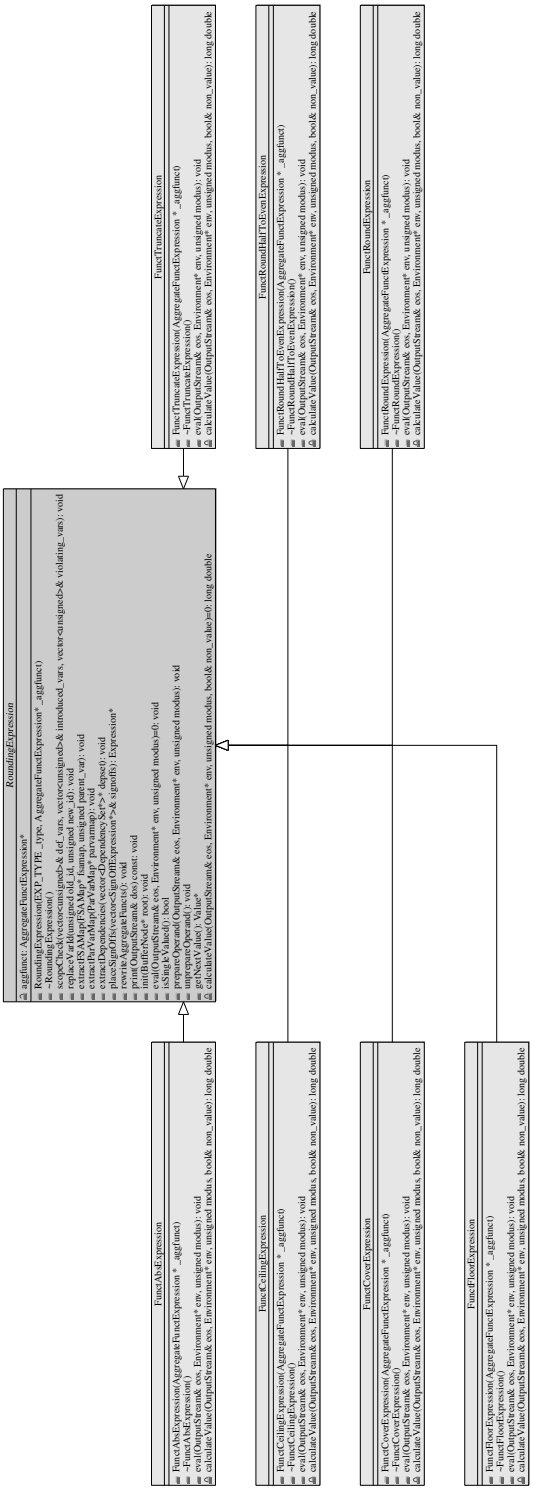
5 Class Diagram: Path Steps



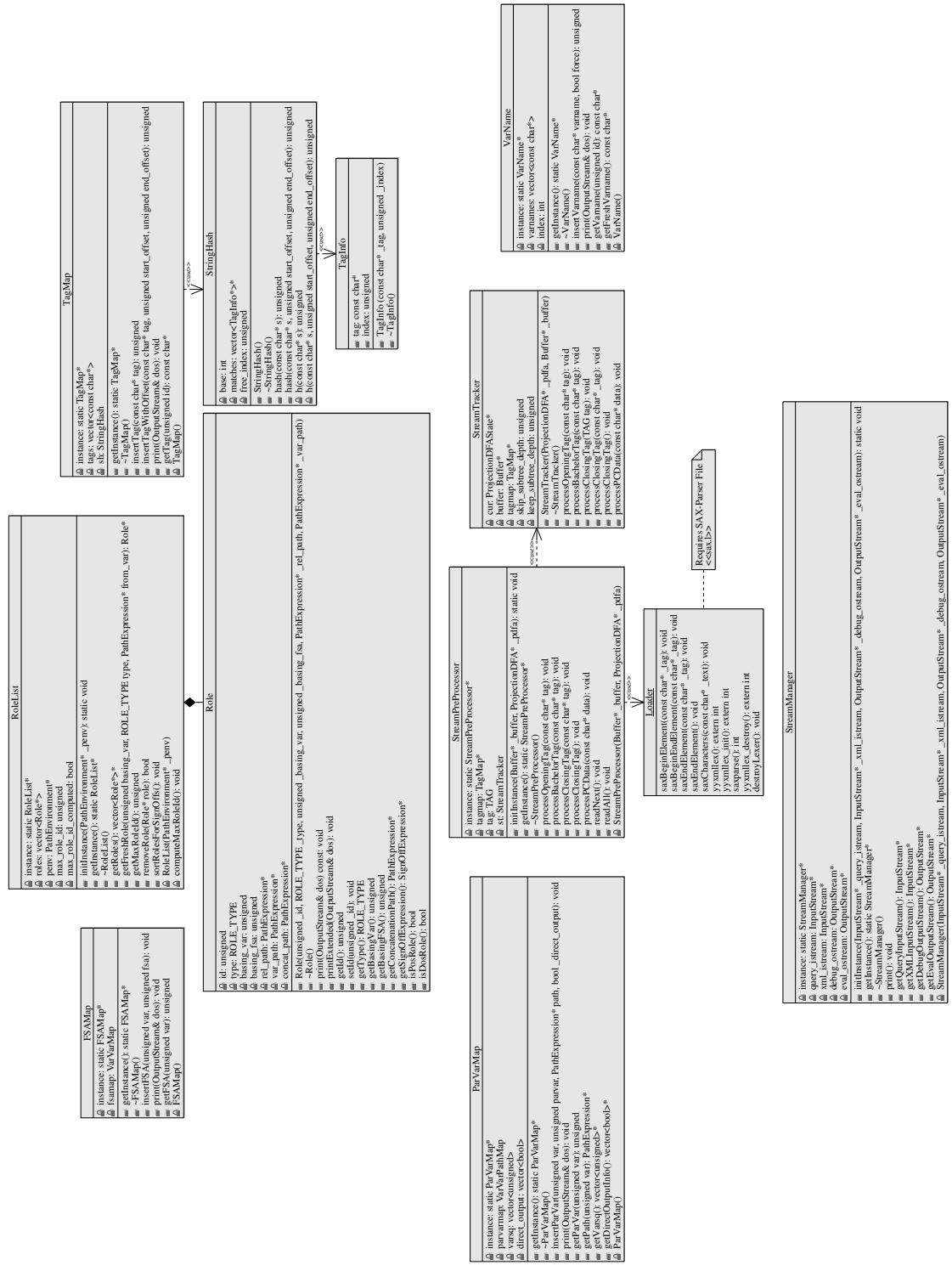
6 Class Diagram: Aggregate Functions



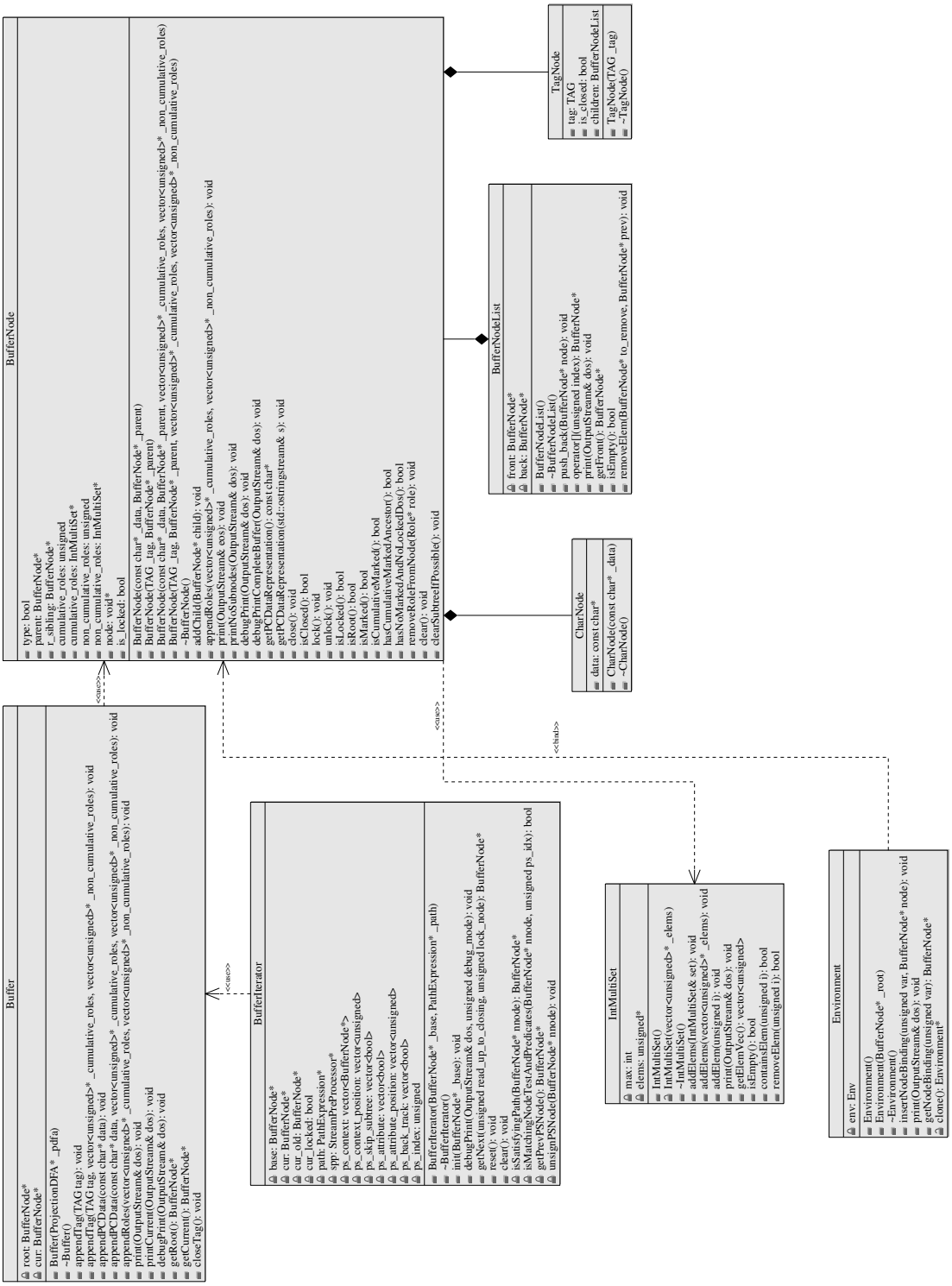
7 Class Diagram: Rounding Functions



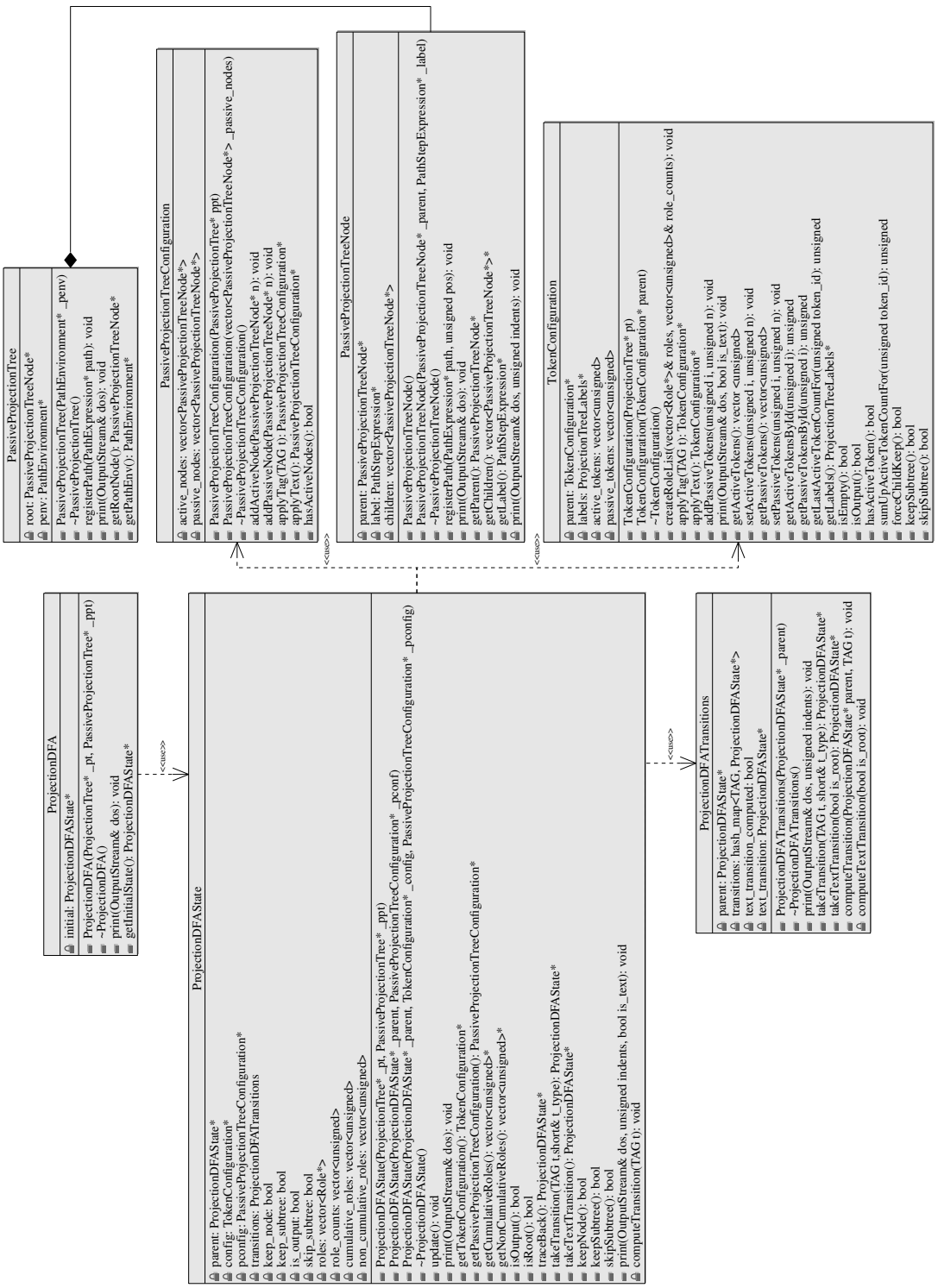
8 Class Diagram: Miscellaneous (Singleton Patterns)



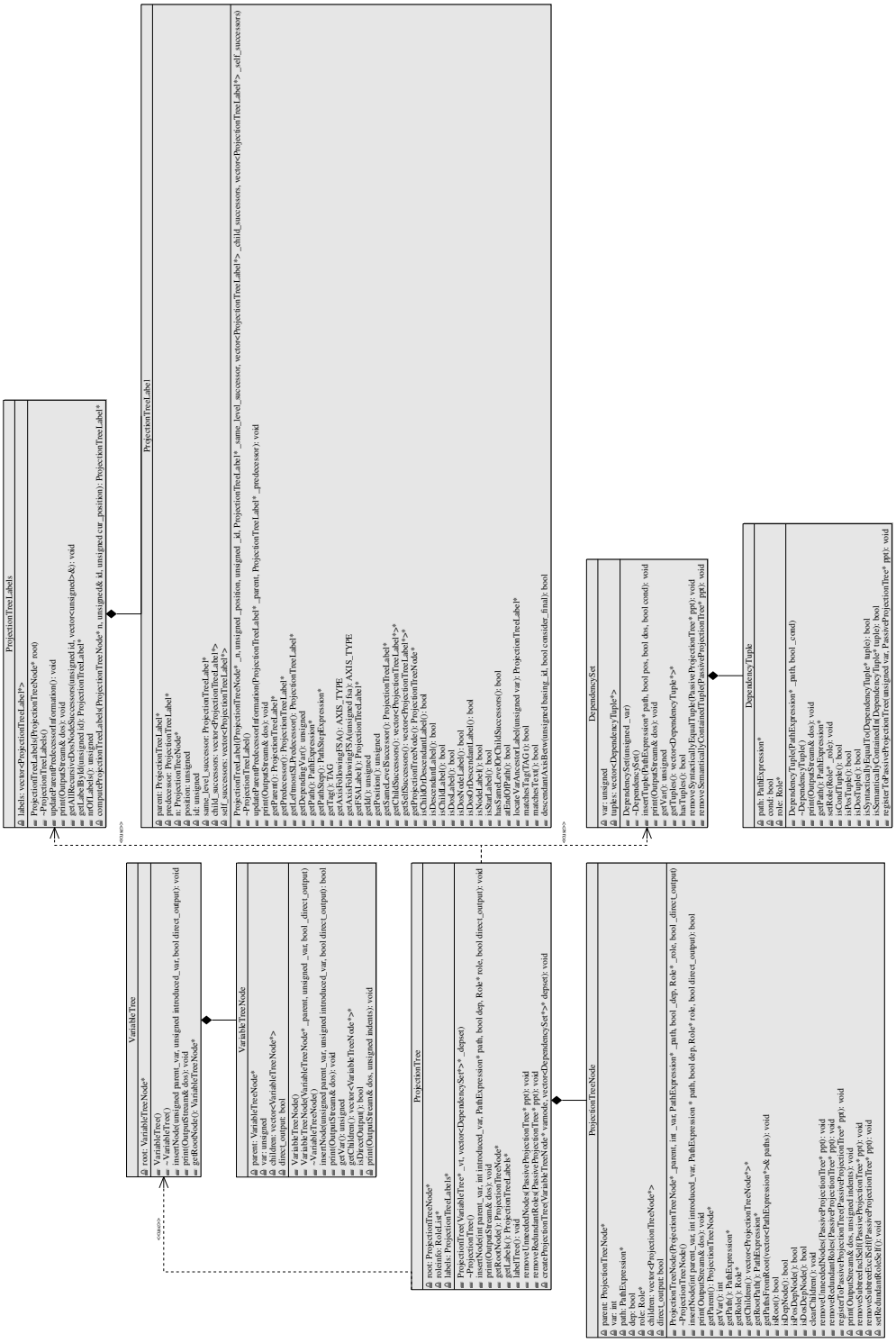
9 Class Diagram: Miscellaneous (Buffer)



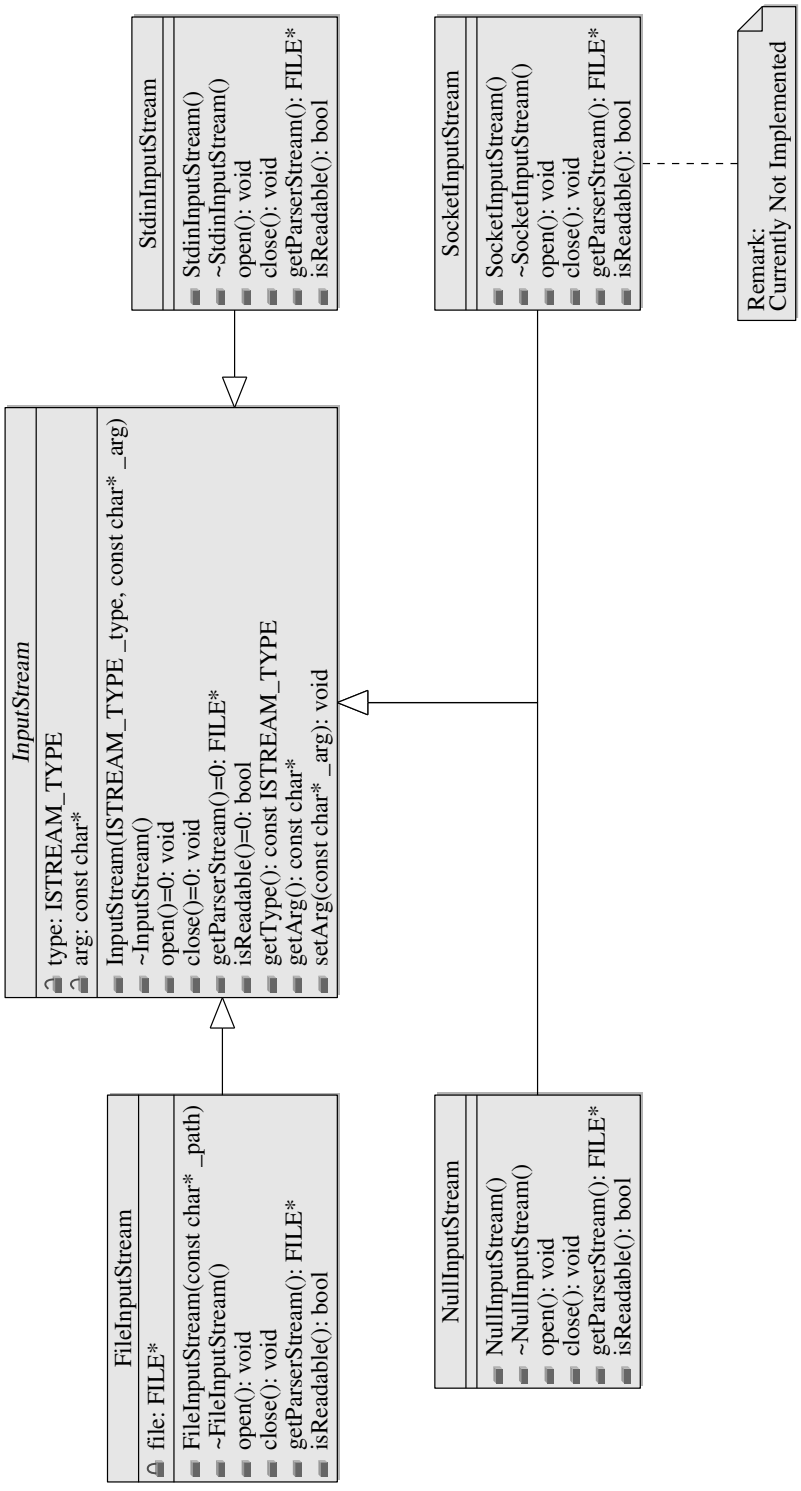
10 Class Diagram: Miscellaneous (Projection)



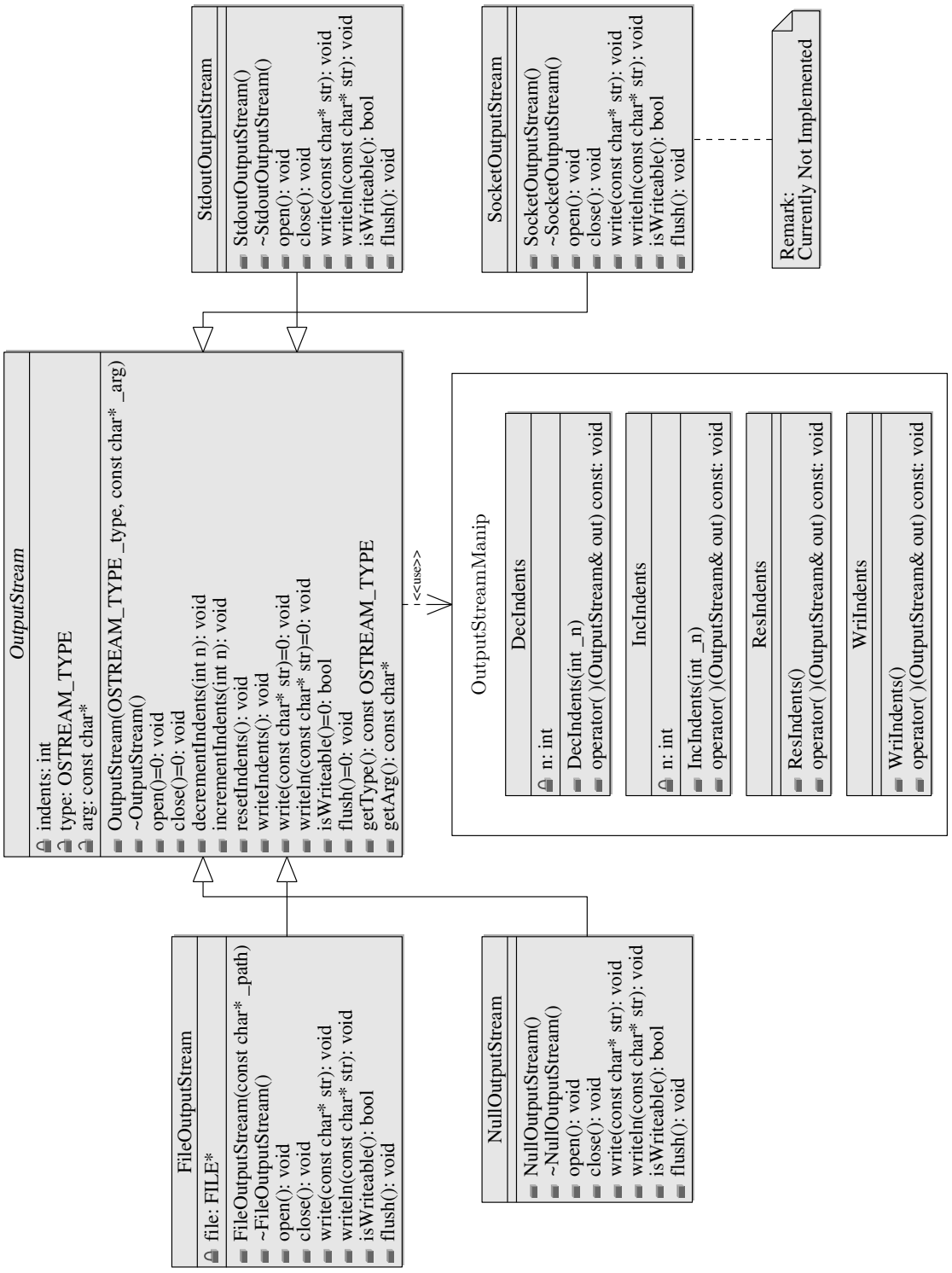
11 Class Diagram: Miscellaneous (Projection (cont'd))



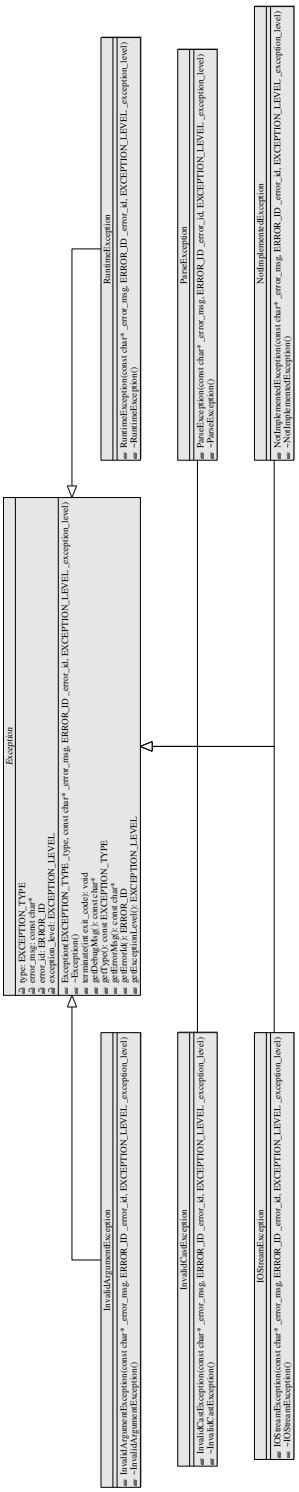
12 Class Diagram: Input Streams



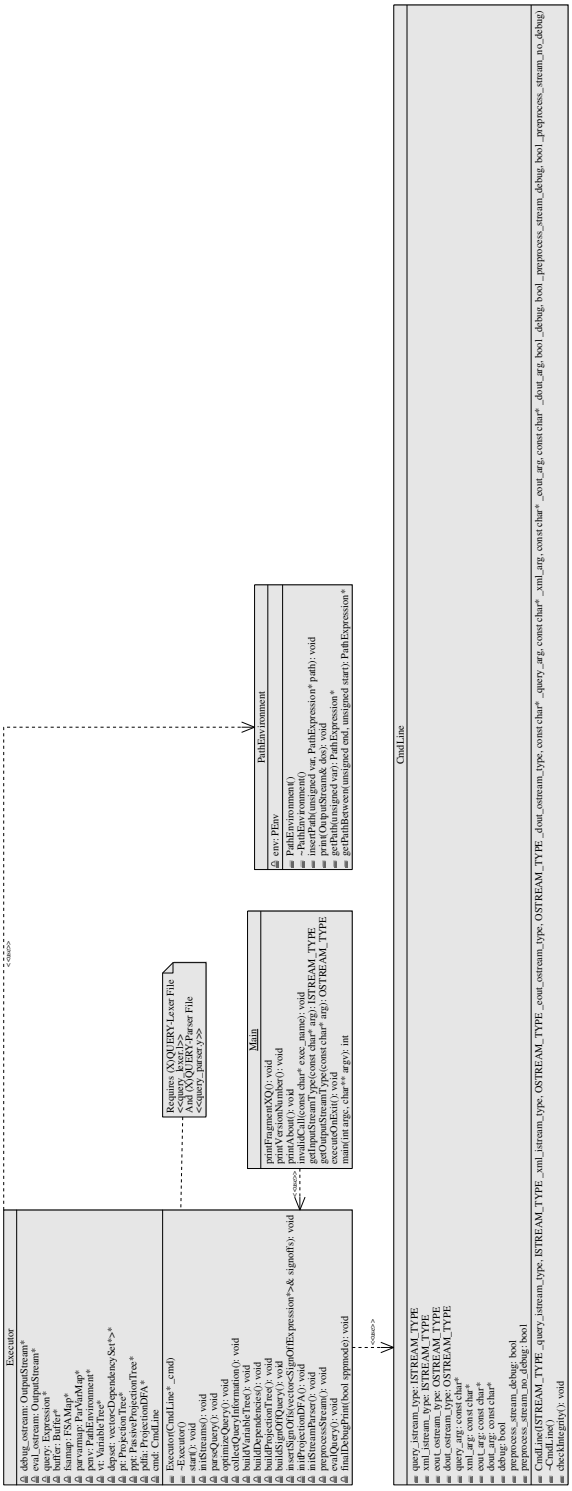
13 Class Diagram: Output Streams



14 Class Diagram: Exceptions



15 Class Diagram: Main/Executor



16 Defines/Enums/Typedef

16.1 Defines

Table 1: Defines

Name	Values	File
OUTPUT_AVG_ON_EMPTY_SEQUENCE	""	aggregatefunctavgexpression.h
OUTPUT_COUNT_ON_EMPTY_SEQUENCE	0	aggregatefunctcountexpression.h
OUTPUT_MAX_ON_EMPTY_SEQUENCE	""	aggregatefunctmaxexpression.h
OUTPUT_MEDIAN_ON_EMPTY_SEQUENCE	""	aggregatefunctmedianexpression.h
OUTPUT_MIN_ON_EMPTY_SEQUENCE	""	aggregatefunctminexpression.h
OUTPUT_STDDEVPOP_ON_EMPTY_SEQUENCE	""	aggregatefunctstddevpopexpression.h
OUTPUT_STDDEVSAMP_ON_EMPTY_SEQUENCE	""	aggregatefunctstddevsampexpression.h
OUTPUT_SUM_ON_EMPTY_SEQUENCE	0	aggregatefunctsumexpression.h
OUTPUT_VARPOP_ON_EMPTY_SEQUENCE	""	aggregatefunctvarpopexpression.h
OUTPUT_VARSAMP_ON_EMPTY_SEQUENCE	""	aggregatefunctvarsampexpression.h
READ_UP_TO_CLOSE_BASE	0	bufferiterator.h
READ_UP_TO_CLOSE_CONTEXT	1	bufferiterator.h
READ_UP_TO_CLOSE_NONE	2	bufferiterator.h
LOCK_CONTEXT_NO_CLEAR	10	bufferiterator.h
LOCK_CONTEXT_LAST_CLEAR	11	bufferiterator.h
LOCK_CONTEXT_ALWAYS_CLEAR	12	bufferiterator.h
LOCK_NONE_NO_CLEAR	13	bufferiterator.h
BIT_DEBUG_MODE_FULL	0	bufferiterator.h
BIT_DEBUG_MODE_PARTIAL	1	bufferiterator.h
BIT_DEBUG_MODE_SIMPLE	2	bufferiterator.h
TYPE_TAG	true	buffernode.h
TYPE_PCDATA	false	buffernode.h
DBG_YACC		debug.h
EVAL_QUERY	0	expression.h
EVAL_QUERY_SILENT	1	expression.h
EVAL_SIGNOFF	2	expression.h
OUTPUT_ABS_ON_NON_VALUE	""	functabsexpression.h
OUTPUT_CEILING_ON_NON_VALUE	""	functceilingexpression.h
OUTPUT_COVER_ON_NON_VALUE	""	functcoverexpression.h
OUTPUT_FLOOR_ON_NON_VALUE	""	functfloorexpression.h
OUTPUT_ROUND_ON_NON_VALUE	""	functroundexpression.h
OUTPUT_ROUNDHALFTOEVEN_ON_NON_VALUE	""	functroundhalftoevenexpression.h
OUTPUT_TRUNCATE_ON_NON_VALUE	""	functtruncateexpression.h
ROUND_TO_DECIMAL_PLACE	16	miscfunctions.h
REPLACE_DECIMAL_POINT_BEFORE	true	miscfunctions.h
REPLACE_DECIMAL_POINT_AFTER	false	miscfunctions.h
NEWLINE	"\n"	outputstream.h
INDENT_TOKEN	" "	outputstream.h
INDENT_MULTIPLICATOR	2	outputstream.h
INDENT_SINGLE	1	outputstream.h
INDENT_DOUBLE	2	outputstream.h
TAG_SHADOW_FRONT	".shadow_front"	pathexpressionadornment.h
TAG_SHADOW_TAIL	".shadow_tail"	pathexpressionadornment.h
WEIGHT_AXIS_CHILD	1	pathstepexpression.h
WEIGHT_AXIS_DESCENDANT	3	pathstepexpression.h
WEIGHT_AXIS_DOS	3	pathstepexpression.h
WEIGHT_INNER_NODETEST	1	pathstepexpression.h
WEIGHT_NODETEST_NODE	3	pathstepnodeexpression.h
WEIGHT_NODETEST_STAR	2	pathstepstarexpression.h
WEIGHT_NODETEST_TAG	2	pathsteptagexpression.h
WEIGHT_NODETEST_TEXT	4	pathsteptextexpression.h

continuation on the next side ...

Table 1: Defines (cont'd)

continuation from previous side ...

Name	Values	File
TRANSITION_UNKNOWN	0	projectiondfatransitions.h
TRANSITION_REGULAR	1	projectiondfatransitions.h
TRANSITION_SKIP_SUBTREE	2	projectiondfatransitions.h
TRANSITION_KEEP_SUBTREE	3	projectiondfatransitions.h
YYDEBUG	1	query_parser.y
YYPRINT(file, type, value)	–	query_parser.y
HASH_BASE	31	stringhash.h
ROOTVAR	".root"	typeenums.h
TAGID_ROOT	0	typeenums.h
VERSION_NUMBER	"2.1"	version.h

16.2 Enums

Table 2: Enums

Name	Values	File
short_opts	opt_iqstream = '1' opt_ixstream = '2' opt_oestream = '3' opt_odstream = '4' opt_query_arg = 'q' opt_xml_arg = 'x' opt_eout_arg = 'e' opt_dout_arg = 'o' opt_debug = 'd' opt_streamdebug = 's' opt_streamnodebug = 'b' opt_fragmentxq = 'f' opt_version = 'v' opt_about = 'a' opt_help = 'h'	main.cpp
ROLE_TYPE	rt_root rt_variable rt_condition rt_output	typeenums.h
AXIS_TYPE	at_child at_descendant at_dos	typeenums.h
NODETEST_TYPE	ntt_tag ntt_star ntt_node ntt_text	typeenums.h
ATTRIBUTE_TYPE	at_position	typeenums.h
COMP_TYPE	ct_lt ct_leq ct_eq ct_gt ct_geq ct_neq	typeenums.h

continuation on the next side ...

Table 2: Enums (cont'd)

continuation from previous side ...

Name	Values	File
EXP_TYPE	ect_and	typeenums.h
	ect_or	
	ect_not	
	ect_exists	
	ect_empty	
	ect_varstep	
	ect_operand	
	ect_true	
	ect_false	
	et_empty	
	et_stringconst	
	et_numericconst	
	et_for	
	et_where	
	et_if	
	et_doc	
	et_comment	
	et_nodeconstr	
	et_path	
	et_pathsteptag	
	et_pathstepstar	
	et_pathstepnode	
	et_pathsteptext	
	et_sequence	
	et_var	
	et_varstep	
	et_signoff	
	et_operandvar	
	et_operandvarstep	
	et_operandaggregatefunct	
	et_operandrounding	
	et_operandconst	
	et_aggregatefunctsum	
	et_aggregatefunctavg	
	et_aggregatefunctmin	
	et_aggregatefunctmax	
	et_aggregatefunctcount	
	et_aggregatefunctstddevsamp	
	et_aggregatefunctstddevpop	
	et_aggregatefunctvarsamp	
	et_aggregatefunctvarpop	
	et_aggregatefunctmedian	
	et_abs	
	et_ceiling	
	et_cover	
	et_floor	
	et_round	
	et_roundhalftoeven	
	et_truncate	
ISTREAM_TYPE	it_file	typeenums.h
	it_null	
	it_socket	
	it_stdin	
OSTREAM_TYPE	ot_file	typeenums.h
	ot_null	
	ot_socket	
	ot_stdout	

continuation on the next side ...

Table 2: Enums (cont'd)

continuation from previous side . . .

Name	Values	File
EXCEPTION_TYPE	exct_argument exct_iostream exct_parse exct_runtime exct_cast exct_impl	typeenums.h
EXCEPTION_LEVEL	excl_warning excl_error excl_fatal	typeenums.h
ERROR_ID	eid_notset = 0 eid_argument = 100 eid_stream = 200 eid_stream_input = 201 eid_stream_output = 202 eid_parse = 300 eid_parse_query = 301 eid_parse_xml = 302 eid_runtime = 400 eid_runtime_illegalmode = 401 eid_runtime_bit = 402 eid_runtime_tokenconfig = 403 eid_runtime_ptlabel = 404 eid_cast = 500 eid_cast_stringnumeric = 501 eid_cast_numericstring = 502 eid_cast_summation = 503 eid_cast_power = 504 eid_cast_radical = 505 eid_cast_division = 506 eid_impl = 600	typeenums.h
VALUE_TYPE	xsd_numeric xsd_string xsd_unknown	typeenums.h

16.3 Typedef

Table 3: Typedef

Name	Values	File
Env	map<unsigned, BufferNode*>	environment.h
VarVarMap	map<unsigned, unsigned>	fsamap.h
VarVarPathMap	map<unsigned, pair<unsigned, PathExpression*> >	parvarmap.h
PEnv	map<unsigned, PathExpression*>	pathenvironment.h
var_list_type	list< pair<VarExpression*, Expression*> >	query_lexer.l
var_list_type	list< pair<VarExpression*, Expression*> >	query_parser.y
TAG	unsigned	tagmap.h